

Rosa Maria Benito ·
Chantal Cherifi · Hocine Cherifi ·
Esteban Moro · Luis M. Rocha ·
Marta Sales-Pardo *Editors*

Complex Networks & Their Applications X

Volume 2, Proceedings of the Tenth
International Conference on Complex
Networks and Their Applications
COMPLEX NETWORKS 2021

Studies in Computational Intelligence

Volume 1073

Series Editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland

The series “Studies in Computational Intelligence” (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

Indexed by SCOPUS, DBLP, WTI Frankfurt eG, zbMATH, SCImago.

All books published in the series are submitted for consideration in Web of Science.

More information about this series at <https://link.springer.com/bookseries/7092>

Rosa Maria Benito · Chantal Cherifi ·
Hocine Cherifi · Esteban Moro ·
Luis M. Rocha · Marta Sales-Pardo
Editors

Complex Networks & Their Applications X

Volume 2, Proceedings of the Tenth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2021



Springer

Editors

Rosa María Benito
Escuela Técnica Superior de Ingeniería
Agronómica, Alimentaria y de Biosistemas
Universidad Politécnica de Madrid
Madrid, Madrid, Spain

Hocine Cherifi
LIB, UFR Sciences et Techniques
University of Burgundy
Dijon, France

Luis M. Rocha
Thomas J. Watson College of Engineering
and Applied Science
Binghamton University
Binghamton, USA

Chantal Cherifi
IUT Lumière
University of Lyon
Bron Cedex, France

Esteban Moro
Grupo Interdisciplinar de
Sistemas Complejos
Universidad Carlos III de Madrid
Leganés, Madrid, Spain

Marta Sales-Pardo
Department of Chemical Engineering
Universitat Rovira i Virgili
Tarragona, Tarragona, Spain

ISSN 1860-949X

ISSN 1860-9503 (electronic)

Studies in Computational Intelligence

ISBN 978-3-030-93412-5

ISBN 978-3-030-93413-2 (eBook)

<https://doi.org/10.1007/978-3-030-93413-2>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2022, corrected publication 2022

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

It is the tenth edition of the “International Conference on Complex Networks & their Applications”, one of the major international events in network science. Every year, it brings together researchers from a wide variety of scientific backgrounds ranging from finance, medicine and neuroscience, biology and earth sciences, sociology and politics, computer science and physics, and many others to review the field’s current state and formulate new directions. The great diversity of the attendees is an opportunity for cross-fertilization between fundamental issues and innovative applications.

The proceedings of this edition, hosted by the Polytechnic University of Madrid in Spain from November 30 to December 02, 2021, contains a good sample of high-quality contributions to the multiple issues of complex networks research.

This edition attracted authors from all over the world, with 424 submissions from 56 countries. Each submission has been peer-reviewed by at least three independent reviewers from the international program committee. The 137 papers included in the proceedings are the results of this rigorous selection process. The challenges for a successful edition are undoubtedly related to the quality of the contributors. The success also goes to the fascinating plenary lectures of the keynote speakers.

- Marc Barthélémy (CEA, France): “Challenges in Spatial Networks”
- Ginestra Bianconi (Queen Mary University of London, UK): “Higher-Order Networks and their Dynamics”
- João Gama (University of Porto, Portugal): “Mining Evolving Large-Scale Networks”
- Dirk Helbing (ETH Zürich, Switzerland): “How Networks Can Change Everything for Better or for Worse”
- Yizhou Sun (UCLA, USA): “Graph-based Neural ODEs for Learning Dynamical Systems”
- Alessandro Vespignani (Northeastern University, USA): “Computational Epidemiology at the time of COVID-19”

Our thanks also go to the speakers of the traditional tutorial sessions for delivering insightful talks on November 29, 2021.

- Elisabeth Lex (Graz University of Technology, Austria) and Markus Schedl (Johannes Kepler University, Austria): “Psychology-informed Recommender Systems”
- Giovanni PETRI (ISI Foundation, Italy): “A crash-course in TDA and higher-order dynamics”

This edition during a pandemic has been quite challenging. The deep involvement of many people, institutions, and sponsors is the key to its success.

We sincerely gratify the advisory board members, Jon Crowcroft (University of Cambridge), Raissa D’Souza (University of California, Davis, USA), Eugene Stanley (Boston University, USA), and Ben Y. Zhao (University of Chicago, USA), for inspiring the essence of the conference.

We record our thanks to our fellow members of the organizing committee: José Fernando Mendes (University of Aveiro, Portugal), Jesús Gomez Gardeñes (University of Zaragoza, Spain), and Huijuan Wang (TU Delft, Netherlands), the lightning sessions chairs, Manuel Marques Pita (Universidade Lusófona, Portugal), José Javier Ramasco (IFISC, Spain), and Taha Yasseri (University of Oxford, UK), the poster sessions chairs, Luca Maria Aiello (ITU Copenhagen, Denmark) and Leto Peel (Université Catholique de Louvain, Belgium) the tutorial chairs, Sabrina Gaito (University of Milan, Italy) and Javier Galeano (Universidad Politécnica de Madrid, Spain) the satellite chairs, Benjamin Renoust (Osaka University, Japan), Xiangjie Kong (Dalian University of Technology, China), the publicity chairs, Regino Criado (Universidad Rey Juan Carlos, Spain) and Roberto Interdonato (CIRAD - UMR TETIS, Montpellier, France) the sponsor chairs. Our profound thanks go to Matteo Zignani (University of Milan, Italy), publication chair, for the tremendous work at managing the submission system and the proceedings publication process. Thanks to Stephany Rajeh (University of Burgundy, France), Web chair, for maintaining the website.

We would also like to record our appreciation for the work of the local committee chair, Juan Carlos Losada (Universidad Politécnica de Madrid, Spain), and all the local committee members, David Camacho (UPM, Spain), Fabio Revuelta (UPM, Spain), Juan Manuel Pastor (UPM, Spain), Francisco Prieto (UPM, Spain), Leticia Perez Sienes (UPM, Spain), Jacobo Aguirre (CSIC, Spain), Julia Martinez-Atienza (UPM, Spain), for their work in managing the sessions. They intensely participated to the success of this edition.

We are also indebted to our partners, Alessandro Fellegara and Alessandro Egro from Tribe Communication, for their passion and patience in designing the conference’s visual identity.

We would like to express our gratitude to our partner journals involved in sponsoring keynote talks: Applied Network Science, EPJ Data Science, Social Network Analysis and Mining, and Entropy.

We are thankful to all those who have contributed to the success of this meeting. Sincere thanks to the authors for their creativity. Finally, we would like to express our most sincere thanks to the program committee members for their considerable efforts in producing high-quality reviews in a minimal time.

These volumes make the most advanced contribution of the international community to the research issues surrounding the fascinating world of complex networks. Their breath, quality, and novelty demonstrate the profound contribution of complex networks in understanding our world. We hope you will enjoy reading the papers as much as we enjoyed organizing the conference and putting this collection of articles together.

Rosa M. Benito
Hocine Cherifi
Esteban Moro
Chantal Cherifi
Luis Mateus Rocha
Marta Sales-Pardo

The original version of the book was revised: The volume of the book has been changed from 1016 to 1073. The correction to the book is available at https://doi.org/10.1007/978-3-030-93413-2_67

Organization and Committees

General Chairs

Rosa M. Benito	Universidad Politécnica de Madrid, Spain
Hocine Cherifi	University of Burgundy, France
Esteban Moro	Universidad Carlos III, Spain

Advisory Board

Jon Crowcroft	University of Cambridge, UK
Raissa D'Souza	Univ. of California, Davis, USA
Eugene Stanley	Boston University, USA
Ben Y. Zhao	University of Chicago, USA

Program Chairs

Chantal Cherifi	University of Lyon, France
Luis M. Rocha	Binghamton University, USA
Marta Sales-Pardo	Universitat Rovira i Virgili, Spain

Satellite Chairs

Sabrina Gaito	University of Milan, Italy
Javier Galeano	Universidad Politécnica de Madrid, Spain

Lightning Chairs

José Fernando Mendes	University of Aveiro, Portugal
Jesús Gomez Gardeñes	University of Zaragoza, Spain
Huijuan Wang	TU Delft, Netherlands

Poster Chairs

Manuel Marques-Pita
José Javier Ramasco
Taha Yasseri

University Lusófona, Portugal
IFISC, Spain
University of Oxford, UK

Publicity Chairs

Benjamin Renoust
Andreia Sofia Teixeira
Michael Schaub
Xiangjie Kong

Osaka University, Japan
University of Lisbon, Portugal
MIT, USA
Dalian University of Technology, China

Tutorial Chairs

Luca Maria Aiello
Leto Peel

Nokia-Bell Labs, UK
UC Louvain, Belgium

Sponsor Chairs

Roberto Interdonato
Regino Criado

CIRAD - UMR TETIS, France
Universidad Rey Juan Carlos, Spain

Local Committee Chair

Juan Carlos Losada

Universidad Politécnica de Madrid, Spain

Local Committee

Jacobo Aguirre
David Camacho
Julia Martínez-Atienza
Juan Manuel Pastor
Leticia Pérez Siñes
Francisco Prieto
Fabio Revuelta

CSIC, Spain
UPM, Spain
UPM, Spain
UPM, Spain
UPM, Spain
UPM, Spain
UPM, Spain

Publication Chair

Matteo Zignani

University of Milan, Italy

Web Chair

Stephany Rajeh

University of Burgundy, France

Program Committee

Jacobo Aguirre	Centro de Astrobiología, Spain
Amreen Ahmad	Jamia Millia Islamia, India
Masaki Aida	Tokyo Metropolitan University, Japan
Luca Maria Aiello	IT University of Copenhagen, Denmark
Marco Aiello	University of Stuttgart, Germany
Esra Akbas	Oklahoma State University, USA
Mehmet Aktas	University of Central Oklahoma, USA
Tatsuya Akutsu	Kyoto University, Japan
Reka Albert	The Pennsylvania State University, USA
Laura Alessandretti	ENS Lyon, Italy
Antoine Allard	Université Laval, Canada
Aleksandra Aloric	Institute of Physics Belgrade, Serbia
Claudio Altafini	Linköping university, Sweden
Benjamin Althouse	Institute for Disease Modeling, USA
Luiz G.A. Alves	Northwestern University, USA
G. Ambika	Indian Institute of Science Education and Research, India
Fred Amblard	University Toulouse 1, France
Andre Franceschi De Angelis	Unicamp, Brazil
Marco Tulio Angulo	National Autonomous University of Mexico, Mexico
Alberto Antonioni	Carlos III University of Madrid, Spain
Nino Antulov-Fantulin	ETH Zurich, Switzerland
Nuno Araujo	Universidade de Lisboa, Portugal
Elsa Arcaute	University College London, UK
Saul Ares	Centro Nacional de Biotecnología, Spain
Panos Argyarakis	Aristotle University of Thessaloniki, Greece
Malbor Asllani	University College Dublin, Ireland
Tomaso Aste	University College London, UK
Martin Atzmueller	Osnabrueck University, Germany
Konstantin Avrachenkov	INRIA, France
Giacomo Baggio	University of Padova, Italy
Rodolfo Baggio	Bocconi University and Tomsk Polytechnic University, Italy
Franco Bagnoli	University of Florence, Italy
Annalisa Barla	Università di Genova, Italy
Paolo Barucca	University College London, UK
Nikita Basov	St. Petersburg State University, Russia
Giulia Bassignana	INSERM, France
Gareth Baxter	University of Aveiro, Portugal
Marya Bazzi	University of Oxford and University of Warwick, UK
Andras A. Benczur	Hungarian Academy of Sciences, Hungary

Rosa M. Benito	Universidad Politécnica de Madrid, Spain
Kamal Berahmand	Queensland University of Technology, Australia
Marc Bertin	Université Claude Bernard Lyon 1, France
Ginestra Bianconi	Queen Mary University of London, UK
Ofer Biham	The Hebrew University of Jerusalem, Israel
Hanjo Boekhout	Leiden University, Netherlands
Johan Bollen	Indiana University Bloomington, USA
Anthony Bonato	Ryerson University, Canada
Christian Bongiorno	Università degli studi di Palermo, Italy
Anton Borg	Blekinge Institute of Technology, Sweden
Javier Borge-Holthoefer	Internet Interdisciplinary Institute, Spain
Stefan Bornholdt	Universität Bremen, Germany
Javier Borondo	UPM, Spain
Cecile Bothorel	IMT Atlantique, France
Federico Botta	The University of Warwick, UK
Julien Bourgeois	UBFC, France
Alexandre Bovet	Mathematical Institute, University of Oxford, UK
Dan Braha	NECSI, USA
Ulrik Brandes	ETH Zürich, Switzerland
Rion Brattig Correia	Instituto Gulbenkian de Ciência, Portugal
Markus Brede	University of Southampton, UK
Piotr Bródka	Wroclaw University of Science and Technology, Poland
Iulia Martina Bulai	University of Basilicata, Italy
Javier M. Buldu	Center for Biomedical Technology, Spain
Francesco Bullo	UCSB, USA
Raffaella Burioni	Università di Parma, Italy
Noah Burrell	University of Michigan, USA
Fabio Caccioli	University College London, UK
Rajmonda Sulo Caceres	Massachusetts Institute of Technology, USA
Carmela Calabrese	University of Naples Federico, Italy
M Abdullah Canbaz	Indiana University Kokomo, USA
Carlo Vittorio Cannistraci	TU Dresden, Germany
Vincenza Carchiolo	Università di Catania, Italy
Giona Casiraghi	ETH Zürich, Switzerland
Douglas Castilho	Federal University of Minas Gerais, Brazil
Costanza Catalano	Central Bank of Italy, Italy
Remy Cazabet	Université Lyon 1, France
Po-An Chen	National Chiao Tung University, Taiwan
Xihui Chen	University of Luxembourg, Luxembourg
Chantal Cherifi	Lyon 2 University, France
Hocine Cherifi	University of Burgundy, France
Guanghua Chi	Facebook, USA
Peter Chin	Boston University, USA
Matteo Chinazzi	Northeastern University, USA

Matteo Cinelli	University of Venice, Italy
Richard Clegg	Queen Mary University of London, UK
Reuven Cohen	Bar-Ilan University, Israel
Alessio Conte	University of Pisa, Italy
Michele Coscia	IT University of Copenhagen, Denmark
Christophe Crespelle	Université Claude Bernard Lyon 1, France
Regino Criado	Universidad Rey Juan Carlos, Spain
Pascal Crépey	EHESP, France
Mihai Cucuringu	University of Oxford, USA
Marcelo Cunha	Federal Institute of Bahia, Brazil
Bhaskar Dasgupta	University of Illinois at Chicago, USA
Joern Davidsen	University of Calgary, Canada
Toby Davies	University College London, UK
Pasquale De Meo	Vrije Universiteit Amsterdam, Italy
Fabrizio De Vico Fallani	Inria - ICM, France
Guillaume Deffuant	Cemagref, France
Charo I. del Genio	Coventry University, UK
Pietro Delellis	University of Naples Federico II, Italy
Yong Deng	Xi'an Jiaotong University, China
Mathieu Desroches	Inria Sophia Antipolis Méditerranée Research Centre, France
Patrick Desrosiers	Université Laval, Canada
Karel Devriendt	University of Oxford, UK
Kuntal Dey	Accenture Tech Labs, India
Riccardo Di Clemente	University of Exeter, UK
Matías Di Muro	Universidad Nacional de Mar del Plata, Argentina
Constantine Dovrolis	Georgia Institute of Technology, USA
Ahlem Drif	Sétif 1 University, Algeria
Johan Dubbeldam	Delft University of Technology, Netherlands
Jordi Duch	Universitat Rovira i Virgili, Spain
Nicolas Dugué	LIUM, France
Victor M Eguiluz	IFISC, Spain
Frank Emmert-Streib	Tampere University of Technology, Finland
Alexandre Evsukoff	COPPE/UFRJ, Brazil
Mauro Faccin	Université de Paris, France
Giorgio Fagiolo	Sant'Anna School of Advanced Studies, Italy
Guilherme Ferraz de Arruda	ISI Foundation, Italy
Daniel Figueiredo	UFRJ, Brazil
Marco Fiore	IMDEA Networks Institute, Spain
Alessandro Flammini	Indiana University Bloomington, USA
Manuel Foerster	Bielefeld University, Germany
Angelo Furno	University of Lyon, France
Sabrina Gaito	University of Milan, Italy
Javier Galeano	Universidad Politécnica de Madrid, Spain

Lazaros Gallos	Rutgers University, USA
Riccardo Gallotti	Fondazione Bruno Kessler, Italy
José Manuel Galán	Universidad de Burgos, Spain
Joao Gama	University of Porto, Portugal
Yerali Gandica	Cergy Paris Université, France
Jianxi Gao	Rensselaer Polytechnic Institute, USA
Floriana Gargiulo	University of Paris Sorbonne, France
Alexander Gates	Northeastern University, USA
Vincent Gauthier	Telecom Sud Paris, France
Ralucca Gera	Naval Postgraduate School, USA
Raji Ghawi	Technical University of Munich, Germany
Tommaso Gili	IMT School for Advanced Studies Lucca, Italy
Silvia Giordano	SUPSI, Switzerland
David Gleich	Purdue University, USA
Antonia Godoy	University College London, Spain
Kwang-Il Goh	Korea University, South Korea
Jesus Gomez-Gardenes	Universidad de Zaragoza, Spain
Antonio Gonzalez-Pardo	Universidad Rey Juan Carlos, Spain
Bruno Gonçalves	Data for Science, Inc., USA
Joana Gonçalves-Sá	Nova School of Business and Economics, Portugal
Przemyslaw Grabowicz	University of Massachusetts, Amherst, USA
Carlos Gracia-Lázaro	Institute for Biocomputation and Physics of Complex Systems, Spain
Jean-Loup Guillaume	Université de la Rochelle, France
Mehmet Gunes	Stevens Institute of Technology, USA
Sergio Gómez	Universitat Rovira i Virgili, Spain
Meesoon Ha	Chosun University, South Korea
Jürgen Hackl	University of Liverpool, Switzerland
Aric Hagberg	Los Alamos National Laboratory, USA
Chris Hankin	Imperial College London, UK
Yukio Hayashi	Japan Advanced Institute of Science and Technology, Japan
Mark Heimann	Lawrence Livermore National Laboratory, USA
Torsten Heinrich	Chemnitz University of Technology, Germany
Denis Helic	Graz University of Technology, Austria
Chittaranjan Hens	Indian Statistical Institute, India
Samuel Heroy	University College London, UK
Takayuki Hiraoka	Aalto University, Finland
Philipp Hoevel	University College Cork, Ireland
Petter Holme	Tokyo Institute of Technology, Japan
Seok-Hee Hong	University of Sydney, Australia
Ulrich Hoppe	University Duisburg-Essen, Germany
Yanqing Hu	Sun Yat-sen Univ., China
Sylvie Huet	Inrae, France

Yuichi Ikeda	Kyoto University, Japan
Roberto Interdonato	CIRAD - UMR TETIS, France
Antonio Iovanella	University of International Studies of Rome, Italy
Gerardo Iñiguez	Central European University, Austria
Mahdi Jalili	RMIT University, Australia
Jaroslaw Jankowski	West Pomeranian University of Technology, Poland
Marco Alberto Javarone	Centro Ricerche Enrico Fermi, UCL - CBT, Italy
Hawoong Jeong	Korea Advanced Institute of Science and Technology, South Korea
Tao Jia	Southwest University, China
Chunheng Jiang	Rensselaer Polytechnic Institute, USA
Jiaojiao Jiang	University of New South Wales, Australia
Ming Jiang	University of Illinois at Urbana-Champaign, USA
Di Jin	University of Michigan, USA
Hang-Hyun Jo	The Catholic University of Korea, South Korea
Ivan Jokić	Delft University of Technology, Netherlands
Jason Jung	Chung-Ang University, South Korea
Marko Jusup	Tokyo Institute of Technology, Japan
Arkadiusz Jędrzejewski	Wrocław University of Science and Technology, Poland
Rushed Kanawati	Université Sorbonne Paris Nord, France
Eytan Katzav	The Hebrew University of Jerusalem, Israel
Mehmet Kaya	Firat University, Turkey
Domokos Kelen	Institute for Computer Science and Control, Hungary
Dror Kenett	Johns Hopkins University, USA
Mohammad Khansari	University of Tehran, Iran
Khaldoun Khashanah	Stevens Institute of Technology, USA
Hamamache Kheddouci	Université Claude Bernard, France
Hyounghick Kim	Sungkyunkwan University, South Korea
Jinseok Kim	University of Michigan, USA
Maksim Kitsak	Northeastern University, USA
Mikko Kivela	Aalto University, Finland
Konstantin Klemm	IFISC, Spain
Peter Klimek	Medical University of Vienna, Austria
Andrei Klishin	University of Pennsylvania, USA
Dániel Kondor	SMART, Singapore
Xiangjie Kong	Zhejiang University of Technology, China
Onerva Korhonen	Aalto University/Universidad Politécnica de Madrid, Finland
Elka Korutcheva	Universidad Nacional de Educación a Distancia, Spain
Dimitris Kotzinos	Paris University, France
Reimer Kuehn	King's College London, UK

Prosenjit Kundu	State University of New York at Buffalo, USA
Ryszard Kutner	University of Warsaw, Poland
Haewoon Kwak	Singapore Management University, Singapore
Richard La	University of Maryland, USA
José Lages	Université Bourgogne Franche-Comté, France
Renaud Lambiotte	University of Oxford, UK
Aniello Lampo	Universitat Oberta de Catalunya, Spain
Valentina Lanza	Université du Havre, France
Paul Laurienti	Wake Forest, USA
Anna T. Lawniczak	University of Guelph, Canada
Eric Leclercq	University of Burgundy - Laboratory of Informatics (LIB), France
Deok-Sun Lee	Korea Institute for Advanced Study, South Korea
Sune Lehmann	Technical University of Denmark, Denmark
Balazs Lengyel	Hungarian Academy of Sciences, Hungary
Maxime Lenormand	INRAE, France
Juergen Lerner	University of Konstanz, Germany
Claire Lesieur	CNRS, France
Haiko Lietz	GESIS, Germany
Fabrizio Lillo	University of Bologna, Italy
Ji Liu	Stony Brook University, USA
Run-ran Liu	Hangzhou Normal University, China
Giacomo Livan	University College London, UK
Lorenzo Livi	University of Manitoba, Canada
Juan Carlos Losada	Universidad Politécnica de Madrid, Spain
Meilian Lu	Beijing University of Posts and Telecommunication, China
Luca Luceri	SUPSI, Switzerland
John C.S. Lui	The Chinese University of Hong Kong, Hong Kong
Leonardo Maccari	University of Venice, Italy
Matteo Magnani	Uppsala University, Sweden
Cécile Mailler	UVSQ, France
Fragkiskos Malliaros	Paris-Saclay University, France
Giuseppe Mangioni	University of Catania, Italy
Rosario Nunzio Mantegna	Palermo University, Italy
Madhav Marathe	University of Virginia, USA
Manuel Sebastian Mariani	University of Zurich, Switzerland
Radek Marik	Czech Technical University, Czechia
Daniele Marinazzo	Ghent University, Belgium
Andrea Marino	University of Florence, Italy
Antonio Marques	Universidad Rey Juan Carlos, Spain
Manuel Marques-Pita	Universidade Lusofona, Portugal
Christoph Martin	Hamburg University of Applied Sciences, Germany

Cristina Masoller	Universitat Politècnica de Catalunya, Spain
Rossana Mastrandrea	IMT Institute of Advanced Studies, Italy
Ruth Mateos de Cabo	Universidad CEU San Pablo, Spain
Michael Mathioudakis	University of Helsinki, Finland
John Matta	Southern Illinois University Edwardsville, USA
Fintan McGee	Luxembourg Institute of Science and Technology, Luxembourg
Matúš Medo	Bern University Hospital and University of Bern, Switzerland
Jose Fernando Mendes	University of Aveiro, Portugal
Ronaldo Menezes	University of Exeter, UK
Humphrey Mensah	Syracuse University, USA
Engelbert Mephu Nguifo	University Clermont Auvergne - LIMOS - CNRS, France
Anke Meyer-Baese	FSU, USA
Salvatore Micciche	University of Padova, Italy
Radosław Michalski	Wrocław University of Science and Technology, Poland
Letizia Milli	CNR Pisa, Italy
Andreea Minca	Cornell University, USA
Boris Mirkin	Higher School of Economics Moscow, UK
Shubhangshu Mishra	University of Illinois at Urbana-Champaign, USA
Lewis Mitchell	The University of Adelaide, Australia
Bivas Mitra	Indian Institute of Technology Kharagpur, India
Marija Mitrović Dankulov	Institute of Physics Belgrade, Serbia
Andrzej Mizera	University of Luxembourg, Luxembourg
Osnat Mokry	University of Haifa, Israel
Roland Molontay	Budapest University of Technology and Economics, Hungary
Raul Mondragon	Queen Mary University of London, UK
Misael Mongiovì	Consiglio Nazionale delle Ricerche, Italy
Alfredo Morales	MIT Media Lab, USA
Andres Moreira	Universidad Tecnica Federico Santa Maria, Chile
Esteban Moro	Universidad Carlos III de Madrid, Spain
Greg Morrison	University of Houston, USA
Sotiris Moschoyiannis	University of Surrey, UK
Igor Mozetič	Jozef Stefan Institute, Slovenia
Peter Mucha	Dartmouth College, USA
Alberto P. Munuzuri	Univ. of Santiago de Compostela, Sudan
Masayuki Murata	Osaka University, Japan
Tsuyoshi Murata	Tokyo Institute of Technology, Japan
Matthieu Nadini	City, University of London, UK
Alexandre Nicolas	Université Claude Bernard Lyon 1, France
Peter Niemeyer	Leuphana Universität Lüneburg, Germany
Jordi Nin	Universitat Ramon Llull, Spain

Rogier Noldus	Ericsson, Netherlands
El Faouzi Nour-Eddin	Université Gustave Eiffel, France
Neave O'Clery	University College London, UK
Masaki Ogura	Osaka University, Japan
Marcos Oliveira	University of Exeter/GESIS, USA
Andrea Omicini	Alma Mater Studiorum–Università di Bologna, Italy
Gergely Palla	Statistical and Biological Physics Research Group of HAS, Hungary
Pietro Panzarasa	Queen Mary University of London, UK
Fragkiskos Papadopoulos	Cyprus University of Technology, Cyprus
Symeon Papadopoulos	Information Technologies Institute, Greece
Michela Papandrea	SUPSI, Switzerland
Philip Pare	Purdue University, USA
Han Woo Park	Yeungnam University, South Korea
Juyong Park	Korea Advanced Institute of Science and Technology, South Korea
Pierre Parrend	EPITA Strasbourg, France
Leto Peel	Maastricht University, Belgium
Tiago Peixoto	Central European University, and ISI Foundation, Germany
Matjaz Perc	University of Maribor, Slovenia
Anthony Perez	LIFO - Université d'Orléans, France
Lilia Perfeito	LIP - Laboratório de Instrumentação e Física Experimental de Partículas, Portugal
Giovanni Petri	ISI Foundation, Italy
Juergen Pfeffer	Technical University of Munich, Germany
Carlo Piccardi	Politecnico di Milano, Italy
Bruno Pinaud	Université de Bordeaux, France
Flavio Pinheiro	Universidade NOVA de Lisboa, USA
Clara Pizzuti	National Research Council of Italy (CNR), Italy
Pawel Pralat	Ryerson University, Canada
Francisco Prieto-Castrillo	Universidad Politécnica de Madrid, Spain
Christophe Prieur	Telecom ParisTech, France
Natasa Przulj	University College London, Spain
Oriol Pujol	University of Barcelona, Spain
Rami Puzis	Ben Gurion University of the Negev, Israel
Christian Quadri	University of Milan, Italy
Filippo Radicchi	Northwestern University, USA
Jose J. Ramasco	IFISC (CSIC-UIB), Spain
Asha Rao	RMIT University, Australia
Miguel Rebollo	Universitat Politècnica de València, Spain
Gesine Reinert	University of Oxford, UK
Juan Restrepo	University of Colorado Boulder, USA
Daniel Rhoads	Universitat Oberta de Catalunya, Spain

Pedro Ribeiro	University of Porto, Portugal
Laura Ricci	Università di Pisa, Italy
Marian-Andrei Rizoiu	University of Technology, Sydney, Australia
Alessandro Rizzo	Politecnico di Torino, Italy
Luis M. Rocha	Binghamton University, USA
Fernando Rosas	Imperial College London, UK
Giulio Rossetti	KDD Lab ISTI-CNR, Italy
Fabrice Rossi	Université Paris Dauphine, France
Camille Roth	CNRS, Germany
Celine Rozenblat	University of Lausanne Institut de Géographie, Switzerland
Giancarlo Ruffo	Università degli Studi di Torino, Italy
Alex Rutherford	Max Planck Institute, Germany
Marta Sales-Pardo	Universitat Rovira i Virgili, Spain
Iraj Saniee	Bell Labs, Nokia, USA
Francisco C. Santos	Universidade de Lisboa, Portugal
Hiroki Sayama	Binghamton University, USA
Nicolas Schabanel	École Normale Supérieure de Lyon, France
Michael Schaub	RWTH Aachen University, Germany
Maximilian Schich	Prof. for Cultural Data Analytics, Estonia
Frank Schweitzer	ETH Zurich, Switzerland
Hamida Seba	University Lyon1, France
Emre Sefer	Ozyegin University, Turkey
Santiago Segarra	Rice University, USA
Irene Sendiña-Nadal	Rey Juan Carlos University, Spain
Termeh Shafie	University of Manchester, UK
Saray Shai	Wesleyan University, USA
Yilun Shang	Northumbria University, UK
Aneesh Sharma	Google, USA
Rajesh Sharma	University of Tartu, Estonia
Erez Shmueli	Tel-Aviv University, Israel
Julian Sienkiewicz	Warsaw University of Technology, Poland
Anurag Singh	National Institute of Technology Delhi, India
Tiratha Raj Singh	Jaypee University of Information Technology, India
Rishabh Singhal	Dayalbagh Educational Institute, India
Per Sebastian Skardal	Trinity College, USA
Oskar Skibski	University of Warsaw, Poland
Keith Smith	Nottingham Trent University, UK
Igor Smolyarenko	Brunel University, UK
Zbigniew Smoreda	Orange Labs, France
Annalisa Socievole	National Research Council of Italy, Italy
Albert Sole	Universitat Oberta de Catalunya, Spain
Sucheta Soundarajan	Syracuse University, USA
Jaya Sreevalsan-Nair	IIIT Bangalore, India

Massimo Stella	University of Exeter, UK
Fabian Stephany	University of Oxford, UK
Cedric Sueur	Université de Strasbourg, France
Kashin Sugishita	Tokyo Institute of Technology, Japan
Xiaoqian Sun	Beihang University, China
Xiaoqian Sun	Chinese Academy of Sciences, China
Michael Szell	IT University of Copenhagen, Denmark
Bosiljka Tadic	Jozef Stefan Institute, Slovenia
Andrea Tagarelli	University of Calabria, Italy
Patrick TAILLANDIER	INRAE/MIAT, France
Kazuhiro Takemoto	Kyushu Institute of Technology, Japan
Frank Takes	Leiden University, Netherlands
Fabien Tarissan	ENS Paris-Saclay (ISP), France
Ana Maria Tarquis	Universidad Politécnica de Madrid, Spain
Claudio Juan Tessone	Universität Zürich, Switzerland
François Théberge	Tutte Institute for Mathematics and Computing, Canada
I-Hsien Ting	National University of Kaohsiung, Taiwan
Michele Tizzoni	ISI Foundation, Italy
Olivier Togni	Burgundy University, France
Joaquin J. Torres	University of Granada, Spain
Vincent Antonio Traag	Leiden University, Netherlands
Jan Treur	Vrije Universiteit Amsterdam, Netherlands
Janos Török	Budapest University of Technology and Economics, Hungary
Stephen Uzzo	New York Hall of Science, USA
Lucas D. Valdez	IFIMAR, UNMdP-CONICET, Argentina
Pim van der Hoorn	Eindhoven University of Technology, Netherlands
Piet Van Mieghem	Delft University of Technology, Netherlands
Onur Varol	Sabanci University, Turkey
Balazs Vedres	University of Oxford, UK
Wouter Vermeer	Northwestern University, USA
Christian Lyngby Vestergaard	Institut Pasteur, France
Nathalie Vialaneix	INRAE, France
Javier Villalba-Diez	Hochschule Heilbronn, Germany
Johannes Wachs	Central European University, Hungary
Huijuan Wang	Delft University of Technology, Netherlands
Lei Wang	Beihang University, China
Ingmar Weber	Qatar Computing Research Institute, Qatar
Guanghui Wen	Southeast University, China
Karoline Wiesner	University of Potsdam, Germany
Gordon Wilfong	Nokia Bell Labs, USA
Mateusz Wilinski	Los Alamos National Laboratory, USA
Richard Wilson	University of York, UK

Dirk Witthaut	Forschungszentrum Jülich, Germany
Bin Wu	Beijing University of Posts and Telecommunications, China
Jinshan Wu	Beijing Normal University, China
Haoxiang Xia	Dalian University of Technology, China
Fei Xiong	Beijing Jiaotong University, China
Xiaoke Xu	Dalian Minzu University, China
Gitanjali Yadav	University of Cambridge, UK
Gang Yan	Tongji University, Shanghai, China, China
Xiaoran Yan	Zhejiang Lab, China
Taha Yasseri	University College Dublin, Ireland
Ying Ye	Nanjing University, China
Sandro Zampieri	University of Padova, Italy
An Zeng	Beijing Normal University, China
Paolo Zeppini	GREDEG Group de Recherche en Droit Economie et Gestions, France
Yuxia Zhang	Beijing Information Science and Technology University, China
Zi-Ke Zhang	Hangzhou Normal University, China
Matteo Zignani	Università degli Studi di Milano, Italy
Eugenio Zimeo	University of Sannio, Italy
Lorenzo Zino	University of Groningen, Netherlands
Fabiana Zollo	Ca' Foscari University of Venice, Italy
Arkaitz Zubiaga	Queen Mary University of London, UK

Contents

Information Spreading in Social Media

Hate Speech Detection on Social Media Using Graph Convolutional Networks	3
Seema Nagar, Sameer Gupta, C. S. Bahushruth, Ferdous Ahmed Barbhuiya, and Kuntal Dey	
Capturing the Spread of Hate on Twitter Using Spreading Activation Models	15
Seema Nagar, Sameer Gupta, Ferdous Ahmed Barbhuiya, and Kuntal Dey	
Indirect Causal Influence of a Single Bot on Opinion Dynamics Through a Simple Recommendation Algorithm	28
Niccolo Pescetelli, Daniel Barkoczi, and Manuel Cebrian	
Modelling the Effects of Self-learning and Social Influence on the Diversity of Knowledge	42
Tuan Pham	
Activator-Inhibitor Model for Describing Interactions Between Fake News and Their Corrections	54
Masaki Aida and Ayako Hashizume	
Love and Hate During Political Campaigns in Social Networks	66
Juan Carlos Losada, José Manuel Robles, Rosa María Benito, and Rafael Caballero	
Homophily - a Driving Factor for Hate Speech on Twitter	78
Seema Nagar, Sameer Gupta, C. S. Bahushruth, Ferdous Ahmed Barbhuiya, and Kuntal Dey	
Influencing the Influencers: Evaluating Person-to-Person Influence on Social Networks Using Granger Causality	89
Richard Kuzma, Iain J. Cruickshank, and Kathleen M. Carley	

How the Far-Right Polarises Twitter: ‘Hashjacking’ as a Disinformation Strategy in Times of COVID-19	100
Philipp Darius and Fabian Stephany	
Models of Influence Spreading on Social Networks	112
Vesa Kuikka and Minh An Antti Pham	
Bubble Effect Induced by Recommendation Systems in a Simple Social Media Model	124
Franco Bagnoli, Guido de Bonfioli Cavalcabo, Benedetto Casu, and Andrea Guazzini	
Maximum Entropy Networks Applied on Twitter Disinformation Datasets	132
Bart De Clerck, Filip Van Utterbeeck, Julien Petit, Ben Lauwens, Wim Mees, and Luis E. C. Rocha	
Community Deception in Networks: Where We Are and Where We Should Go	144
Valeria Fionda and Giuseppe Pirrò	
Mitigating the Backfire Effect Using Pacing and Leading	156
Qi Yang, Khizar Qureshi, and Tauhid Zaman	
Exploring Bias and Information Bubbles in YouTube’s Video Recommendation Networks	166
Baris Kirdemir and Nitin Agarwal	
Diffusion and Epidemics	
Modeling the Spread of COVID-19 Over Varied Contact Networks	181
Ryan Solorzano and Theresa Migler	
Benchmarking Optimal Control for Network Dynamic Systems with Plausible Epidemic Models	194
Yaroslav V. Salii	
Hardness Results for Seeding Complex Contagion with Neighborhoods	207
Meher Chaitanya and Ulrik Brandes	
Influence Maximization in Complex Networks Through Supervised Machine Learning	217
Owais A. Hussain and Faraz Zaidi	
Story of Two Populations in Epidemics: Is Every Infection Counted?	229
Van Sy Mai and Richard J. La	

A Framework for Simulating Multiple Contagions Over Multiple Networks	241
Aparna Kishore, Lucas Machi, Chris J. Kuhlman, Dustin Machi, and S. S. Ravi	
Finding Influential Nodes in Complex Networks Using Nearest Neighborhood Trust Value	253
Koduru Hajarathaiah, Murali Krishna Enduri, and Satish Anamalamudi	
Two-Population SIR Model and Strategies to Reduce Mortality in Pandemics	265
Long Ma, Maksim Kitsak, and Piet Van Mieghem	
Hypergraph Laplacians in Diffusion Framework	277
Mehmet Emin Aktas and Esra Akbas	
Lumping Reductions for Multispread in Multi-Layer Networks	289
Tatjana Petrov and Stefano Tognazzi	
Microscopic Markov Chain Approach for Measuring Mobility Driven SARS-CoV-2 Transmission	301
Trevor G. Kent, Nolan E. Phillips, Ian McCulloh, Viveca Pavon-Harr, and Heather G. Patsolic	
Epidemics in a Synthetic Urban Population with Multiple Levels of Mixing	315
Alessandro Celestini, Francesca Colaiori, Stefano Guarino, Enrico Mastrostefano, and Lena Rebecca Zastrow	
Dynamics on/of Networks	
From Mean-Field to Complex Topologies: Network Effects on the Algorithmic Bias Model	329
Valentina Pansanella, Giulio Rossetti, and Letizia Milli	
Towards Control of Opinion Diversity by Introducing Zealots into a Polarised Social Group	341
Antoine Vendeville, Benjamin Guedj, and Shi Zhou	
Sensing Enhancement on Complex Networks	353
Markus Brede and Guillermo Romero Moreno	
Cascading Failures and the Robustness of Cooperation in a Unified Scale-Free Network Model	365
Mingxuan He, Matthew Gao, Yang Gao, and Fernanda M. Elliott	
Effect of Peer Influence and Looting Concerns on Evacuation Behavior During Natural Disasters	377
Matthew Hancock, Nafisa Halim, Chris J. Kuhlman, Achla Marathe, Pallab Mozumder, S. S. Ravi, and Anil Vullikanti	

Eigenvalues of Random Signed Graphs with Cycles: A Graph-Centered View of the Method of Moments with Practical Applications	390
Pau Vilimelis Aceituno	
Bosonic Random Walk Neural Networks for Graph Learning	401
Shiv Shankar and Don Towsley	
Machine Learning and Networks	
Multiple Role Discovery in Complex Networks	415
Shu Liu, Fujio Toriumi, Mao Nishiguchi, and Shohei Usui	
Graph Summarization with Latent Variable Probabilistic Models	428
Shintaro Fukushima, Ryoga Kanai, and Kenji Yamanishi	
Image Keypoint Matching Using Graph Neural Networks	441
Nancy Xu, Giannis Nikolentzos, Michalis Vazirgiannis, and Henrik Boström	
Relations Between Entropy and Accuracy Trends in Complex Artificial Neural Networks	452
Lucia Cavallaro, Marco Grassia, Giacomo Fiumara, Giuseppe Mangioni, Pasquale De Meo, Vincenza Carchiolo, Ovidiu Bagdasar, and Antonio Liotta	
Curved Markov Chain Monte Carlo for Network Learning	461
John Sigbeku, Emil Saucan, and Anthea Monod	
Extracting Semantic Information from Dynamic Graphs of Geometric Data	474
Devavrat Vivek Dabke and Bernard Chazelle	
Graph-based Retrieval for Claim Verification over Cross-document Evidence	486
Misael Mongiovì and Aldo Gangemi	
Supervised Link Weight Prediction Using Node Metadata	496
Larissa Mori, Mario Ventresca, and Toyya A. Pujol	
Graph Embedding	
Joint Use of Node Attributes and Proximity for Node Classification	511
Arpit Merchant and Michael Mathioudakis	
Online Updates of Knowledge Graph Embedding	523
Luo Fei, Tianxing Wu, and Arif Khan	
High-Speed and Noise-Robust Embedding of Hypergraphs Based on Double-Centred Incidence Matrix	536
Shuta Ito and Takayasu Fushimi	

Simple Negative Sampling for Link Prediction in Knowledge Graphs	549
Md Kamrul Islam, Sabeur Aridhi, and Malika Smail-Tabbone	
Evaluation of Anomaly Detection for Cybersecurity Using Inductive Node Embedding with Convolutional Graph Neural Networks	563
Amani Abou Rida, Rabih Amhaz, and Pierre Parrend	
Hybrid Graph Embedding Techniques in Estimated Time of Arrival Task	575
Vadim Porvatov, Natalia Semenova, and Andrey Chertok	
What Is Learned in Knowledge Graph Embeddings?	587
Michael R. Douglas, Michael Simkin, Omri Ben-Eliezer, Tianqi Wu, Peter Chin, Trung V. Dang, and Andrew Wood	
Biological Networks	
CUBCO: Prediction of Protein Complexes Based on Min-cut Network Partitioning into Biclique Spanned Subgraphs	605
Sara Omranian and Zoran Nikolic	
Quantifying Cellular Pluripotency and Pathway Robustness Through Forman-Ricci Curvature	616
Kevin A. Murgas, Emil Saucan, and Romeil Sandhu	
Emergence of Stable Functional Cliques in Developing Neural Networks	629
Myles Akin and Yixin Guo	
Complex Networks Reveal Biological Functions of START Domains in Rice: Insights from Computational Systems Biology	641
Sanjeet Kumar Mahtha, Citu, Abhishek Prasad, and Gitanjali Yadav	
Supervised Gene Function Prediction Using Spectral Clustering on Gene Co-expression Networks	652
Miguel Romero, Óscar Ramírez, Jorge Finke, and Camilo Rocha	
Analysis of the San Luis Obispo Bay Microbiome from a Network Perspective	664
Viet Nguyen, Paul Anderson, Alexis Pasulka, and Theresa Migler	
From Quantitative SBML Models to Boolean Networks	676
Athénaïs Vaginay, Taha Boukhobza, and Malika Smail-Tabbone	
Can Dynamic Functional Connectivity Be Used to Distinguish Between Resting-State and Motor Imagery in EEG-BCIs?	688
Paula G. Rodrigues, Carlos A. Stefano Filho, André K. Takahata, Ricardo Suyama, Romis Attux, Gabriela Castellano, João R. Sato, Slawomir J. Nasuto, and Diogo C. Soriano	

Network Medicine

- Inferred Networks and the Social Determinants of Health** 703
Prashant Sanjel and John Matta

- Network-Based Analysis of Prescription Opioids Dispensing
Using Exponential Random Graph Models (ERGMs)** 716
Hilary Aroke, Natallia Katenka, Stephen Kogut, and Ashley Buchanan

- Drug Repositioning Using Multiplex-Heterogeneous Network
Embedding: A Case Study on SARS-CoV2** 731
Léo Pio-Lopez

- Drug Repurposing Using Link Prediction on Knowledge Graphs
with Applications to Non-volatile Memory** 742
Sarel Cohen, Moshik Hershcovitch, Martin Taraz, Otto Kißig,
Andrew Wood, Daniel Waddington, Peter Chin, and Tobias Friedrich

Urban Systems and Networks

- Understanding Imbalance Mechanisms in Shared Mobility Systems** 757
Francisco Prieto-Castrillo, Rosa M. Benito, and Javier Borondo

- Neighborhood Discovery via Network Community Structure** 769
Aaron Bramson

- CityChrono: an Interactive Platform for Transport Network Analysis
and Planning in Urban Systems** 780
Indaco Biazzo

- VLT-LUT: Modeling the Very Long-Term Evolution of the City
in 300 Years** 792
Ariel Castillo, Francisco Martínez, Pedro Donoso, Leonel Gutiérrez,
and Ricardo de la Paz Guala

- Layered Hodge Decomposition for Urban Transit Networks** 804
Unchitta Kan and Eduardo López

- Correction to: Complex Networks & Their Applications X** C1
Rosa Maria Benito, Chantal Cherifi, Hocine Cherifi, Esteban Moro,
Luis M. Rocha, and Marta Sales-Pardo

- Author Index** 817

Information Spreading in Social Media



Hate Speech Detection on Social Media Using Graph Convolutional Networks

Seema Nagar¹(✉), Sameer Gupta², C. S. Bahushruth³,
Ferdous Ahmed Barbhuiya¹, and Kuntal Dey¹

¹ Institute of Information Technology, Guwahati, Guwahati, India
seema.nagar@iiitg.ac.in

² National Institute of Technology, Kurukshetra, Kurukshetra, India
³ Manipal University, Jaipur, India

Abstract. Detection of hateful content on Twitter has become the need of the hour. Hate detection is challenging primarily due to the subjective definition of “hateful”. In the absence of context, text content alone is often not sufficient to detect hate. In this paper, we propose a framework that combines content with context, to detect hate. The framework takes into account (a) textual features of the content and (b) unified features of the author to detect hateful content. We use a Variational Graph Auto-encoder (VGAE) to jointly learn the unified features of authors using a social network, content produced by the authors, and their profile information. To accommodate emerging and future language models, we develop a flexible framework that incorporates any text encoder as a plug-in to obtain the textual features of the content. We empirically demonstrate the performance and utility of the framework on two diverse datasets from Twitter.

Keywords: Hate speech detection · Graph convolutional network

1 Introduction

Recent times have witnessed a significant increase of hate speech on Twitter. The unprecedented growth of hateful tweets on Twitter is inciting real life violence. Therefore, detecting hateful tweets is an utmost important task to the research community. The earlier works [22] utilize manually crafted textual features detect hateful content. With the advent of deep learning, automatic feature selection from text has become more prevalent [1,3,23]. A school of researchers argue that textual features are not sufficient to detect hate. [6,20] showed the effectiveness of combining user features and context with text in order to improve hate detection.

On social media platforms, a user is influenced by his or her neighbours. Therefore, [4,20] demonstrated the importance of manually crafted features from the social network structure in detecting hateful content. However, in practice, which social network features are important and should be considered in the hate speech detection task is a laborious and time consuming process. Therefore, a

feature vector which can encode a user’s position in a social network structure is desirable. Prior works [15, 17] suggest that the evolving nature of language and the personalized hateful meaning of benign words or phrases in a community is crucial in detecting hateful content. In summary, hate detection model should factor in, a) textual features, b) social network structure features (graph features) and c) user features in term of the language used or any other meta-data of the users.

In this paper, we propose to learn from all three types of features, wherein the social structure features are auto learned using deep neural networks, as oppose to manually crafting of features. We provide a generic framework for hate detection where any text encoder can be plugged in to encode textual features, along with the unified features of the author/user. The unified features of a user capture three crucial aspects, 1) social network structure, 2) language usage and 3) meta-data. We use variational graph auto-encoder (VGAE) [11] to learn the unified feature of a user. Latent representation obtained for nodes in a graph using VGAE has proved to be effective in link prediction and node classification tasks [10]. VGAE jointly learns from the social network structure, a user’s past tweets and the profile information available. The textual features and VGAE embedding, together help us to capture the contextual, social and the personalized nature of hate.

We make following contributions in this paper:

1. We propose a framework for hate speech detection in a holistic way where any textual encoder can be plugged in to encode textual features of a tweet
2. We construct two feature types for an individual tweet, a) textual features of the tweet and b) unified user features, wherein the unified features captures social network structure, previous tweets and meta-data of a user using a VGAE
3. We empirically demonstrate the effectiveness of our proposed approach on two diverse datasets from Twitter

2 Related Work

Hate speech detection on micro-blogs has received significant attention in the research community. It evolved from using only manually crafted lexical and linguistic features [22] to more sophisticated automatic feature selection using deep neural network. [1, 3, 23] used deep neural network for automatic feature selection. A survey by [19] detailed existing approaches using natural language processing techniques to detect hateful content on social media.

Recently [12] used a Simple Word Embeddings-based model to customize the word embedding to hateful content and showed that it beats state-of-the-art model even with a much simpler model and much lesser parameters. Burnap et al. [2] proposed a data-driven blended model to study intersectionality of different forms of hate, specifically race and sexual orientation and shows it improves the machine classification of cyber hate. Park et al. [16] modelled hate speech detection as a two-step approach using a neural network with two binary

classifiers. In the first step, predict the presence of abusive speech, and in the second step, discern the form of abusive speech.

Kwok et al. [13] argued that in the absence of contextual information, detecting hate from text is very difficult. Recently [7, 8] presented weaknesses of the currently used machine learning approaches for detecting hate. [15, 17] suggested language evolves, and even benign words or phrases are being used to spread hate in closed communities as they associate personalized meaning, make detecting hateful content a challenging task. Vijayaraghavan et al. [21] bridged the gap by using socio-political context to detect different categories of hate and also shows why explainability of models is very important. Gao et al. [6] showed the importance of context in detecting hateful news articles and comments.

Unsvaag et al. [20] shows that incorporating network related user features provided improvements in hate detection. Founta et al. [4] propose a unified architecture incorporating tweet text, author and social network features. They use features such as clustering coefficient, various types of position related metrics (hub, authority, eigenvector and closeness centrality) from follower and followee network.

We realize that a holistic view for hateful content detection is crucial. The detection should factor in the evolving language, personalized meanings of words/phrases in communities, and the larger context in which the hateful content is produced. Therefore, capturing critical signals such as user language and social context is crucial to improving hate detection. Our proposed framework bridges the gap.

3 The Proposed Approach

3.1 Proposed Model Overview

Our proposed model makes use of multi-modal data present on Twitter. In particular, we use two types of modalities, a) text and b) social network structure. We construct two types of features for a tweet t posted by a user u , a) textual features from the text content of the tweet and b) unified user features. The unified user features capture three types of information about a user, a) social network structure, b) language use and c) meta-data. The unified user features are encoded into a fixed size vector using graph convolution networks. Figure 1 shows the proposed framework architecture. The text t is encoded using any text encoder followed by a fully forward neural network (FFNN). The user u 's unified features are encoded by the social network encoder, which is pre-trained separately as described in Sect. 3.2.

3.2 Social Network Encoder as Unified Feature Encoder

The social network encoder captures a user u 's joint latent representation, in the social graph, the language used (behavioural traits) and meta-data. We use a user's past posts for constructing language features, and the profile information

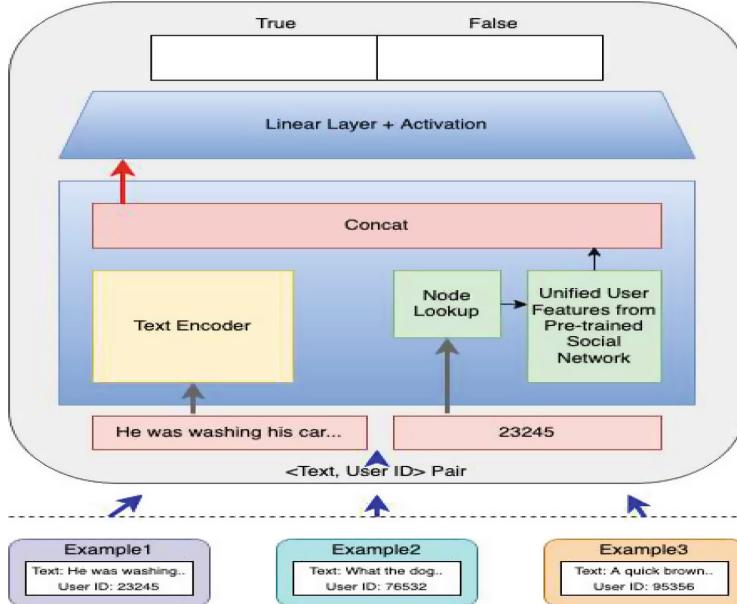


Fig. 1. Model overview

for meta-data features. The social network encoder takes two inputs, i.e. 1) user features and 2) the social connectivity graph. It conducts unsupervised learning to achieve a vector representation of the unified features of a user.

We use a Variational Graph Auto-Encoder (VGAE) [11] to implement our social network encoder. A VGAE consists of two layers of graph convolutional layers as an encoder for encoding a graph into a lower dimensional latent representation Z , followed by a decoder to reconstruct the adjacency matrix by optimizing cross entropy loss between the original and the reconstructed adjacency matrix and KL-divergence of the approximate from the true posterior. Figure 2 shows a schematic representation of VGAE architecture. The social encoder is trained separately. One simple lookup is required to get a user's unified representation (captures social network, language usage and meta features) while training the proposed framework.

Language Usage and Meta-Data Features as User Features. User features are generated from the two types of text a user has produced, a) posts made by the user and b) profile information of the user on Twitter. The mechanism to generate user features is explained below:

1. For each user i , the two types of texts are concatenated in a document d_i
2. Create a corpus D of documents from all the users, where $D = d_1, d_2, \dots, d_n$
3. Each document text is tokenized and tf-idf score is computed for each token

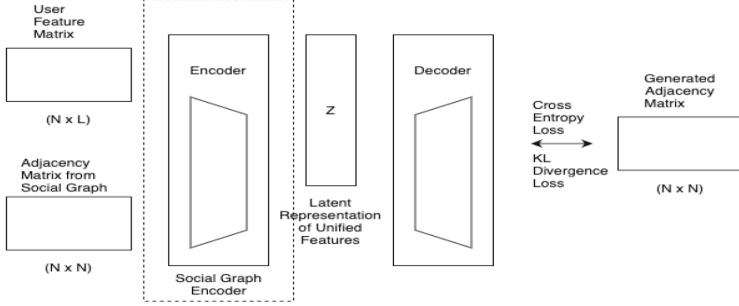


Fig. 2. Variational graph auto encoder

4. Using the tf-idf scores of tokens in d_i , a feature vector v_i is generated for each user i

Variational Graph Auto Encoder (VGAE). The encoder for a VGAE is often written as $q\phi(Z|X, A)$, which takes a data point X and produces a distribution. The distribution is usually parameterized as a multivariate Gaussian. Therefore, the encoder predicts the means and standard deviation of the Gaussian distribution. The lower-dimensional embedding Z is sampled from this distribution. The decoder is a variational approximation, $p\theta(A|Z)$, which takes an embedding Z and produces the output \hat{A} .

Encoder or Inferencing Model

The graph convolution network (GCN) consists of two graph convolutional layers. GCN abstracts the social network graph into a latent representation through a convolutional network. The first layer takes adjacency matrix A and a node feature matrix X as inputs and generates the latent variable Z as output. The first GCN layer generates a lower-dimensional feature matrix \tilde{X} , as shown in the formula in Eq. 1 where $\tilde{A} = D^{-1/2}AD^{-1/2}$ is a symmetrically normalized adjacency matrix. The second GCN layer generates μ and log of σ^2 as shown in Eq. 2 and 3 .

$$\tilde{X} = GCN(X, A) = ReLU(\tilde{A}XW_0) \quad (1)$$

$$\mu = GCN_\mu(X, A) = \tilde{A}\tilde{X}W_1 \quad (2)$$

$$\log\sigma^2 = GCN_\sigma(X, A) = \tilde{A}\tilde{X}W_1 \quad (3)$$

The two layers of GCN can be combined as in Eq. 4 and Z can be calculated using Eq. 5, where $\epsilon \in \mathcal{N}(0, 1)$, wherein \mathcal{N} represents a Gaussian (normal) distribution.

$$GCN(X, A) = \tilde{A}.ReLU(\tilde{A}XW_0)W_1 \quad (4)$$

$$Z = \mu + \sigma * \epsilon \quad (5)$$

In summary, encoder can be written as in Eq. 6.

$$q(z_i|X, A) = N(z_i|\mu_i, \text{diag}(\sigma_i^2)) \quad (6)$$

Decoder

The generative model (decoder) is defined by an inner product between latent variable Z and its transpose Z^T . The output by the decoder (Eq. 8) is reconstructed as adjacency matrix \hat{A} as shown in Eq. 7, where $\sigma()$ is the logistic sigmoid function. In summary, decoder can be represented as in Eq. 8.

$$\hat{A} = \sigma(ZZ^T) \quad (7)$$

$$p(A_{ij} = 1|z_i, z_j) = \sigma(z_i^T z_j) \quad (8)$$

Loss in VGAE

The loss function for variational graph autoencoder has two parts. The first part of the loss function measures how well the network reconstructs the data, and it is called the variational lower bound. We model the loss as binary cross-entropy between the input and the output. The second part of the loss function is the KL-divergence between $q(Z|X, A)$ and $p(Z)$, where $p(Z) = \mathcal{N}(0, 1)$. It measures how closely our $q(Z|X, A)$ matches with $p(Z)$. Formally, the loss can be written as in Eq. 9.

$$L = E_{q(Z|X, A)}[\log p(A|Z)] - KL[q(Z|X, A)||p(Z)] \quad (9)$$

4 Experiments and Results

4.1 Experiments Overview

The purpose of the experiments is the following:

- Compare with the state-of-the-art baselines and investigate the effect of using GCN features in the classification of hateful content.
- Investigate the correlation of graph features with the performance of the proposed framework.
- Investigate the effect of textual length in the performance of the proposed framework.

4.2 Dataset

We use two publicly available *Founta Dataset* [5] and *Ribeiro Dataset* [18] to conduct the experiments. *Founta Dataset* is widely used to benchmark hate detection. This dataset contains four types of labels which are *abusive*, *hateful*, *spam* and *normal*. We discard the tweets labelled as *spam* as they are not needed. We club *hateful* and *abusive* in the class of *hate* to model the problem as a binary

classification problem as the purpose is to show effectiveness of GCN with respect to the stat-of-the-art baselines. We use Twitter API¹ to fetch tweet text, user profile, a user’s previous tweets and follower-followee network.

Ribeiro Dataset contains 200 most recent tweets of 100,386 users, totalling to roughly 19M tweets. It also contains a retweet induced graph of the users. It has 2,286,592 directed edges. The retweet-induced graph is a directed graph $G = (V, E)$ where each node $u \in V$ represents a user in Twitter, and each edge $(u1, u2) \in E$ represents a retweet in the network, where the user $u1$ has retweeted user $u2$. Out of the 100,386 users, labels (hateful or normal) are available for 4,972 users, out of which 544 users are labelled as hateful and the rest as normal. The dataset does not have labels for the tweet content. Therefore, we manually annotate the tweets as hateful or not. Annotating 19M tweets is a very expensive and time-consuming process. Hence we annotate a subset of tweets. We manually annotated 30,720 tweets of a set of users. We pick these users from the set of 4,972 labelled users based on their degree in the retweet network. Every tweet is annotated by three people (having at least one undergrad degree). The majority annotated label is assigned as the final label to a tweet. The dataset is available on request. Table 1 shows the dataset details, reported ignoring the unavailable tweets. We employ pre-processing techniques suitable for tweets. We remove links present, convert emoticons to text and finally remove non-ASCII characters.

Table 1. Dataset description

Dataset	Count of tweets	Count of nodes	Count of edges	% of Hateful tweets
<i>Founta</i>	43,405	47,054	8,11,471	27.5
<i>Ribeiro</i>	30,720	18,642	7,71,401	39.5

4.3 Experiments Setting

User Feature Construction Used by the Social Encoder

To generate user features, we use profile meta-data as well as previous tweets of a user. We use a count-vectorizer² to generate a vector of size 1024. This creates a token score vector for each user. The tokens are created from the tweet texts as well as profile meta-data text. We compute tf-idf score for each token and put the score as value for that token. A token is considered only if it has occurred across meta data of at least 30% users.

¹ <https://developer.twitter.com/en/docs/tweets/post-and-engage/overview>.

² https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html.

Training Variational Graph Auto-Encoder (VGAE)

We use the source code provided by³ to implement VGAE. For every dataset, VGAE is trained separately before using it for the classification task. We use the user features constructed in the manner as mentioned earlier, and the social network graph as input to train VGAE. For training VGAE, the learning rate is 0.01, the number of epochs is 120, the batch size is the whole graph, and Adam optimizer is used. The encoder portion of the trained VGAE is used to encode unified features of a user for further experimentation.

Training the Proposed Framework

We experiment with LSTM as text encoder. In training the framework, we do a simple lookup to the separately trained social graph encoder to get a unified representation of a user. The LSTM encoder consists of a single LSTM layer with an input size of 50 and output size equal to 50. The text encoder is trained during the classifier training. The outputs of the text embedding layer is passed into a feed forward layer that outputs an embedding of size 768.

Textual features and unified features are concatenated to get a joint representation of a tweet, and its author. The joint representation is fed to 2 layers of feed forward neural network, followed by a softmax layer for classification. We use cross entropy loss for the classification task. Training the LSTM based implementation, the learning rate is 0.001, the batch size is 3000, the number of epochs is 30 and the optimizer used is Adam. We use the standard train, test split mechanism for both the datasets and all the baselines, 20% of the tweets are retained for test and remaining for training. In the training set, 10% is retained for validation, using the remaining to train.

4.4 Experimental Results

Comparison with the State-of-the-Art Baselines

Baseline A [4]: We implement [4] wherein textual features as well as the graph features are used. This baseline represents state-of-the-art where graph modality is also used.

Baseline B [9]: This baseline represents state-of-the-art with only using text modality. We remove the GCN based features from the proposed model. This baseline helps us to show that GCN has a significant impact on the performance of the model and textual features are not alone sufficient to detect hate.

We also tried to compare results with the approach proposed by [14]. The paper proposes to use a graph neural network for hate speech detection in a semi-supervised way. However, the approach needs, tweets to be added to the social network graph as nodes. This construct results in a huge graph. With the resources available to us, training VGAE on the datasets is a challenge. Therefore, we could not compare our results with it. Table 2 shows the results of our approach and the two baselines. We see that our approach produces superior

³ https://github.com/rusty1s/pytorch_geometric.

results compared to both the baselines. The gain in the accuracy by adding the social graph encoder features is as high as 6%.

Table 2. Performance of our approach using LSTM based text encoder

	<i>Founta dataset</i>		<i>Ribeiro dataset</i>	
	Accuracy	F1	Accuracy	F1
Our approach	0.7878	0.7586	0.8475	0.8436
Baseline A [4]	0.7295	0.6171	0.8375	0.8318
Baseline B [9]	0.7207	0.6110	0.7962	0.7821

Effect of Tweets Length on Classification Accuracy

Figure 3 shows the accuracy of the proposed framework with respect to the tweet lengths. We observe that for both the datasets and for our approach, as well as for Baseline B, accuracy is highest when the tweet length lies close to 17 and 7, for *Founta Dataset* and *Ribeiro Dataset* respectively. This observation further strengthens the discussion presented in Sect. 5, observing that in very short tweets, context is unavailable. Similarly, for very long tweets, it becomes difficult to extract the dependencies, hence making it difficult for the classifier to detect hate. For *Ribeiro Dataset* this pattern is not so evident since pre-processing removed many un-necessary tokens such as stops words, links, user mentions.

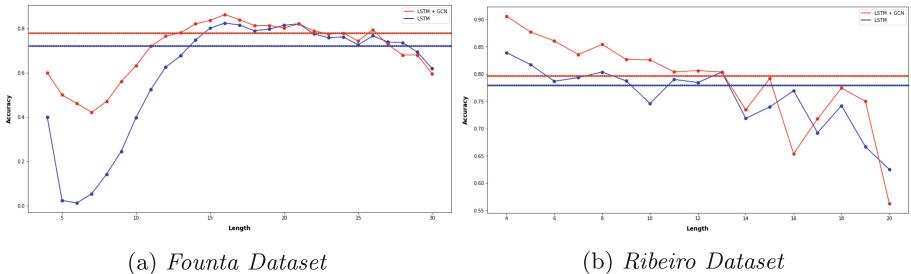


Fig. 3. Plots illustrating effect of text length on classification accuracy. The dotted line shows the overall classifier accuracy

Correlation of Graph Features with Accuracy

Figure 4 illustrates the accuracy of the proposed framework with respect to various graph centrality metrics. Accuracy of the tweets posted by the users in each bucket is computed and plotted. We observe that for both the datasets, centrality values do not correlate with accuracy, which further emphasizes our hypothesis that GCN is required to capture the social structure of a user.

5 Discussion

5.1 Error Analysis

We investigate the errors committed by our system. We manually looked at the predicted labels of a few mis-classified tweets from both the datasets. We notice a few interesting following patterns:

1. Tweets not having enough context around hateful/bad words:
 - “@<USER>” Get your stinking paws off me, you damned dirty ape!” - classified as hateful by our classifier, while originally labelled as normal. This tweet is a quote from the movie “Planet of the Apes”, hence demonstrating how hate depends on the context.
2. Tweets having wrong original annotation by humans:
 - “@<USER> No I destroy my sacroiliac joint. main joint that holds pelvis to sacrum, the strongest ligaments.in body and I rip them apart.” - labelled as hateful, but is normal. The classifier classified it is normal.
3. Tweets needing external context to understand the semantic meaning:
 - “RT @<USER>: How about we fucking hire trans boys to play trans boys <LINK>” - classified as hateful, but is normal. External context is required about the social issue to classify correctly.
4. Tweets where hate is present as sarcasm. An example is the following:
 - “Morning GM! I love my customers just like I love thetoxic free Ingredients in our Locsuria.”.

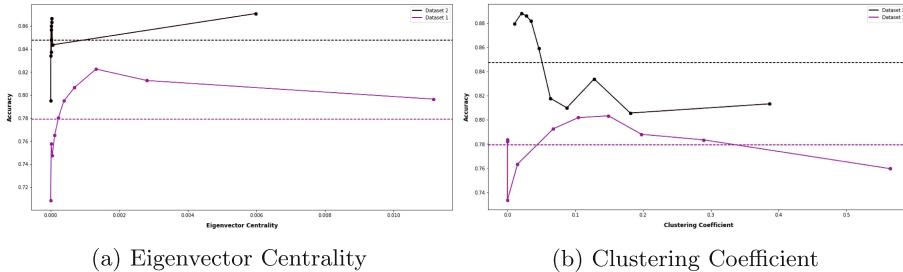


Fig. 4. Plots illustrating correlation of various graph based centrality metrics with the accuracy. The dotted line shows the overall classifier accuracy

5.2 Investigating the Effect of Text Encoders

BERT has also been used recently by researchers, coupled with graph convolution, for citation recommendation [10]. Motivated by this, we had also conducted experiments using a BERT encoder instead of an LSTM encoder. In the process, we fine-tuned the native BERT embedding on our dataset and conducted our experiments.

However, we found the fine-tuned BERT to overfit despite different folding of train/validate data, on both the datasets. We applied our graph convolution network despite the overfit and found improvements that the graph convolution delivered on these two datasets over and beyond the BERT fine-tunings to be around 1% and 0.5% respectively on the two datasets, which appeared to be encouraging at a first-level. The relatively limited size of the data, and the massive number of parameters that a BERT model associates with, is perhaps the most likely reason for such over-fitting.

We note that VGAE uses convolution as it's base mechanism, which is less expensive to train compared to a recurrent neural network mechanism. Also, the number of trainable parameters in the Graph encoder model is only 4,525,056, which is significantly lesser when compared to a typical Recurrent Graph Neural Networks (RecGNNs). We could train the VGAE with much limited resource on google-colab⁴. However, in the future, further work can be done to avoid the BERT-overfitting, and thereby further improve state of the art.

6 Conclusion

We propose a generic framework to detect hateful content. The framework is capable of taking two types of features, a) textual features of the content and b) the author's unified features. Any text encoder can be plugged in to encode the textual features of the content. We use variational graph auto-encoder (VGAE) to capture the unified representation of a user. The unified representation of a user is learnt using VGAE from three types of information, 1) social network, 2) language used in content generated by him and 3) meta-data or profile information. We empirically demonstrate that the proposed framework outperforms the baselines on two diverse datasets from Twitter.

References

1. Badjatiya, P., Gupta, S., Gupta, M., Varma, V.: Deep learning for hate speech detection in tweets. In: WWW, pp. 759–760 (2017)
2. Burnap, P., Williams, M.L.: Us and them: identifying cyber hate on twitter across multiple protected characteristics. EPJ Data Sci. **5**(1), 11 (2016)
3. Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., Bhamidipati, N.: Hate speech detection with comment embeddings. In: WWW, pp. 29–30 (2015)
4. Founta, A.M., Chatzakou, D., Kourtellis, N., Blackburn, J., Vakali, A., Leontiadis, I.: A unified deep learning architecture for abuse detection. In: ACM WebSci, pp. 105–114 (2019)
5. Founta, A.M., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., Vakali, A., Sirivianos, M., Kourtellis, N.: Large scale crowdsourcing and characterization of Twitter abusive behavior. In: ICWSM. AAAI Press (2018)
6. Gao, L., Huang, R.: Detecting online hate speech using context aware models. arXiv preprint [arXiv:1710.07395](https://arxiv.org/abs/1710.07395) (2017)

⁴ <https://colab.research.google.com/>.

7. Gröndahl, T., Pajola, L., Juuti, M., Conti, M., Asokan, N.: All you need is “love” evading hate speech detection. In: Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security, pp. 2–12 (2018)
8. Hosseini, H., Kannan, S., Zhang, B., Poovendran, R.: Deceiving Google’s perspective API built for detecting toxic comments. arXiv preprint [arXiv:1702.08138](https://arxiv.org/abs/1702.08138) (2017)
9. Isaksen, V., Gambäck, B.: Using transfer-based language models to detect hateful and offensive language online. In: Proceedings of the Fourth Workshop on Online Abuse and Harms, pp. 16–27 (2020)
10. Jeong, C., Jang, S., Shin, H., Park, E., Choi, S.: A context-aware citation recommendation model with BERT and graph convolutional networks. arXiv preprint [arXiv:1903.06464](https://arxiv.org/abs/1903.06464) (2019)
11. Kipf, T.N., Welling, M.: Variational graph auto-encoders. arXiv preprint [arXiv:1611.07308](https://arxiv.org/abs/1611.07308) (2016)
12. Kshirsagar, R., Cukuvac, T., McKeown, K., McGregor, S.: Predictive embeddings for hate speech detection on twitter. arXiv preprint [arXiv:1809.10644](https://arxiv.org/abs/1809.10644) (2018)
13. Kwok, I., Wang, Y.: Locate the hate: detecting tweets against blacks. In: Twenty-Seventh AAAI Conference on Artificial Intelligence (2013)
14. Mishra, P., Del Tredici, M., Yannakoudakis, H., Shutova, E.: Abusive language detection with graph convolutional networks. arXiv preprint [arXiv:1904.04073](https://arxiv.org/abs/1904.04073) (2019)
15. Nobata, C., Tetraeault, J., Thomas, A., Mehdad, Y., Chang, Y.: Abusive language detection in online user content. In: Proceedings of the 25th International Conference on World Wide Web, pp. 145–153 (2016)
16. Park, J.H., Fung, P.: One-step and two-step classification for abusive language detection on Twitter. arXiv preprint [arXiv:1706.01206](https://arxiv.org/abs/1706.01206) (2017)
17. Raisi, E., Huang, B.: Cyberbullying identification using participant-vocabulary consistency. arXiv preprint [arXiv:1606.08084](https://arxiv.org/abs/1606.08084) (2016)
18. Ribeiro, M., Calais, P., dos Santos, Y., Almeida, V., Meira Jr, W.: “Like sheep among wolves”: characterizing hateful users on Twitter (2017)
19. Schmidt, A., Wiegand, M.: A survey on hate speech detection using natural language processing. In: Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, pp. 1–10 (2017)
20. Unsväg, E.F., Gambäck, B.: The effects of user features on Twitter hate speech detection. In: Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), pp. 75–85 (2018)
21. Vijayaraghavan, P., Larochelle, H., Roy, D.: Interpretable multi-modal hate speech detection (2013)
22. Warner, W., Hirschberg, J.: Detecting hate speech on the world wide web. In: Second Workshop on Language in Social Media, pp. 19–26. Association for Computational Linguistics (2012)
23. Waseem, Z., Hovy, D.: Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In: NAACL Student Research Workshop, pp. 88–93 (2016)



Capturing the Spread of Hate on Twitter Using Spreading Activation Models

Seema Nagar¹(✉), Sameer Gupta², Ferdous Ahmed Barbhuiya¹,
and Kuntal Dey¹

¹ Institute of Information Technology, Guwahati, Guwahati, India
seema.nagar@iiitg.ac.in

² National Institute of Technology, Kurukshetra, Kurukshetra, India

Abstract. Hate speech is a prevalent and pervasive phenomenon on social media platforms. Detecting hate speech and modelling its spread is a significant research problem with practical implications. Though hate speech detection has been an active area of research, hate spread modelling is still in its nascent stage. Prior works have analyzed the hateful users' social network embedding, hateful user detection using belief propagation models, and the spread velocity of hateful content. However, these prior works fail to factor in the multiple hateful forms (such as hate against gender, race and ethnicity) and the temporal evolution of hate spread, limiting their applicability. We take a holistic approach wherein we model the spread of hate as a single form and fine-granular spread of hateful forms. We extend the traditional spread and activation (SPA) model to capture the spread of hate and its forms. We use SPA to model, the spread of hate as one single form while *TopSPA* captures the spread of multiple hate forms. We also propose ways to detect hateful forms by using latent topics present in hateful content. We empirically demonstrate our approach to a dataset from Twitter that contains ample hate speech instances along with users labelled as hateful or not.

Keywords: Spread and activation · Information spread · Hate speech

1 Introduction

Social media platforms are widely used for disseminating information. Re-posting information lets it flow without any change on social media platforms. Various types of information are shared on Twitter, among which, a common phenomenon is hate speech. There have been many cases where hate speech has incited real-life violence amongst people [8, 10]. Hate speech can be defined as “language that is used to express hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group” [3]. Despite many social media sites and governments’ attempts to curb hate speech, it is still besetting our society. Facebook has been blamed for instigating anti-Muslim mob violence in Sri Lanka and playing a leading role in the possible genocide

of the Rohingya community in Myanmar by spreading hate speech. Therefore, studying the spread of hate is a critical problem to be investigated in detail.

The work on hate spread modelling on social media is limited. The first two studies [7, 10] leverage belief propagation model [5] to detect hateful users in Twitter and Gab respectively. They first find an initial set of hateful users based on hateful lexicons present in the content the users have posted, followed by the belief propagation [5] to find additional hateful users on retweet and re-post induced graphs respectively. Both the works treat the harmfulness of users as a static value that does not change over time. However, we model the action of spreading hate as a factor of a user receiving hate messages and then spreading depending upon her activation level at that time. Thus, we model the hate spread dynamics as a temporally evolving user behaviour, attempting to represent hate spreading dynamics more pragmatically.

Mathew *et al.* [7], further look at the diffusion dynamics of posts generated by hateful vs non-hateful users. They find that posts of hateful users receive a much larger audience and a faster rate. However, they do not look into the spread of multiple topics/hateful forms present in hate speech. It is a crucial aspect as [12] there are differences in spread across topics [12], has not been addressed in the literature. We believe to the best of our knowledge we are the first to address this aspect.

We take a holistic view to capture the spread of hate as one form and the fine-granular spread of different hate forms. We propose to utilize the spreading activation (SPA) model to capture the spread of hate as a single form. We analyze the topics present in the hateful tweets on a Twitter dataset and observe that there are multiple forms of hate present in the tweets and latent topics can be mapped to forms of hate. We believe that the spread of different forms of hate should ideally be captured by a model that considers these different forms. Hence, we modify the existing model called *TopSPA* [4] to capture the hate-related nuances in the *TopSPA* model. We do not compare the results with the linear threshold and independent cascade based methods, since SPA and *TopSPA* outperform them. To summarise, we make the following contributions in this paper:

1. We use *SPA* based model for capturing the spread of hate as a single form.
2. We modify the existing *TopSPA* model to capture the spread of hateful forms.
3. We empirically demonstrate that *SPA* and *TopSPA* based models can outperform a baseline for capturing the spread of hate and different forms of hate on a Twitter dataset.

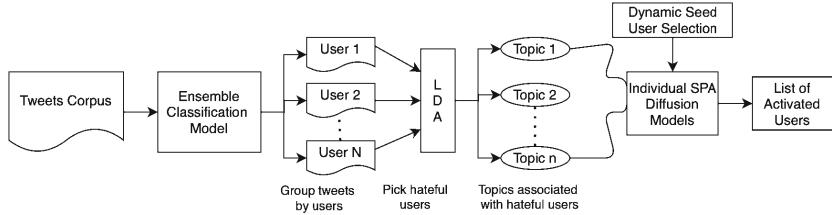


Fig. 1. Proposed architecture

2 Related Work

Information spread modelling on social networks is an active research area [6]. There are many previous papers [13, 16] which have extended traditional epidemic spread models to social media. Xuejun *et al.* [16] extend the traditional epidemic models to have one more status for contacted for better modelling spread of topics on Micro-blogging platforms. Wang *et al.* [14] considers that emotional information present in tweets is an important indicator of information being forwarded by the receiver. They proposed the ESIS model based on the SIS model, where the proportion of forwarded information with an emotional quality is used as an edge weight in the model. It proved that information diffusion is related to propagation probability and transmission intensity, and therefore its performance is better than that of the SIS model.

Ribeiro *et al.* [10] present a first study on the characterization of hateful users in comparison with normal users on Twitter. They find that instead of the general assumption that hateful users are alone, they are actually central in their social network. Their methodology leverages the belief propagation model proposed by [5] to extend an initial set of hateful users. Mathew *et al.* [7] are the first to look into the diffusion dynamics of hateful content on Gab (gab.com) which hosts a large amount of hateful content. They leverage the methodology proposed by [10] to find hateful users. The initial set of hateful users found using a lexicon based approach may miss the users whose posts mainly contain images, videos, or URLs. They use belief propagation [5], suggested in [10] on the re-post network, wherein the belief of initial hateful users is being set to one. Further, they look at the diffusion dynamics of posts generated by hateful vs non-hateful users. They find that hate speech has a much higher spreading velocity. The posts of hateful users receive a much larger audience and as well at a faster rate.

3 Central Idea

In this section, we present our approach for modelling the spread of hate as a single form and along with multiple hateful forms. We use the native spreading and activation (SPA) algorithm to model the spread of hate as a single form, and the modified *TopSPA* to the model spread of different hate forms. We first describe the machine learning model used for classifying the posts as hateful or

not and how we utilize that to classify users as hateful. From the hateful users, we select seed users for SPA and *TopSPA*. SPA being topic agnostic starts with seed users having equal energy. In contrast, in *TopSPA*, a seed user is initialized with an energy vector, where energy for each hateful form is proportional to the user’s affinity towards it. Our approach is summarized in Fig. 1.

3.1 Classification

Tweet Classification. Ensemble-based classification models have performed better than individual classification models [9]. We utilize various data available for hate detection that captures the nuances of hate speech such as toxicity, obscenity, threats, insults, identity hate, racism and sexism.

Two classification models, A and B, are trained independently on two separate datasets. We take the mean of the predictions as the final classification score. Both models follow [15]’s philosophy that the feature vector is computed using Naive Bayes log count ratios, bi-grams and unigrams. However, we use logistic regression instead of support vector machine for classification as it performs better in our experiments. Moreover, we manually annotate 4,720 tweets as hateful or not to validate our approach.

User Classification. A user is characterized by the posts he makes. The posts include his original tweets, retweets or quotes. We compute the hate score of a user u_i as the fraction of the posts classified as hateful. Let the posts made by a user is a set P . Let the posts classified as hateful is a set H . Formally, the score for a user u_i can be computed as follows: $HS(i) = |H|/|P|$. We classify a user as hateful if the score is greater than a threshold Th_{Hate} , which is set empirically.

3.2 The Native SPA Model

The traditional SPA is an iterative process. Given a directed weighted graph with V nodes and E edges, it starts with a few nodes as seed nodes. The seed nodes are initialized with some initial energy and considered as activated nodes. In every iteration, each activated node passes a fraction of its energy to its directed neighbours. The fraction of energy passed to a neighbour is determined by two factors, a) the dissipation factor and b) the edge weight from the node to the neighbouring node. A node is activated when it’s total energy received goes above a Th_{SPA} threshold. We do not allow multiple firing of a node to capture real behaviour, where a hateful user once hateful remains hateful for a while.

The social network graph $G = (V; E)$ is represented as an $N \times N$ adjacency matrix W . W_{ij} denotes the weight of the edge from node i to j . We normalize the adjacency matrix W such that weights of the outgoing edges from node i , sum up to 1. The normalized \hat{W} matrix is obtained as follows:

$$(\forall w_{ij} \text{ in } W), \hat{w}_{ij} = \frac{w_{ij}}{\sum_{j=1}^N w_{ij}} \quad (1)$$

Clearly, $\sum(\hat{w}_{i,1}, \hat{w}_{i,2}, \dots, \hat{w}_{i,n}) = 1$. \hat{W} , the transformed matrix, is subsequently used to distribute energy proportionate to connection strengths along $\Gamma(v)$, the direct neighbours of a given vertex v . The process terminates when either of the following condition meets, a) when no new nodes are activated, b) after a steady activation state, with respect to some delta, is reached, or c) when a maximum number of iterations is exceeded. We use the SPA as it is in modelling the spread of hate as a single form. We pick a fraction of hateful users as seed users and then let the SPA process run.

3.3 Topic Modelling

We use a latent topic detection technique called LDA [1] to detect the latent topics present in a tweet. Due to tweets being short in length and large in number, scaling LDA to detect topics where every tweet is treated as one document is very challenging. Therefore, we create one document per user by concatenating all his posts, including tweets, retweets and quotes. Let a user u_i have made posts P , where $P = p_1, p_2, \dots, p_N$. Then, the document d_i for user u_i is created by concatenating all the tweets in one document. Therefore, we have:

$$d_i = \cup_{(p_j \in P)} p_j \quad (2)$$

Let $D = (\forall i \in 1..n) d_i$ be the corpus of documents. We further investigate D to detect latent topics present in it using LDA based techniques. We explore two variants of sampling for LDA, a) variational Bayes sampling method and b) Gibbs Sampling. Let the set of latent topics is T , where $T = t_1, t_2, \dots, t_n$. The latent topic modelling produces a vector of topic affinity scores v_{d_i} for each document d_i . Let a_i is the affinity scores with respect to topic t_i , then we have topic affinity vector as follows:

$$T_{d_i} = \langle a_1, a_2, a_3, \dots, a_T \rangle \quad (3)$$

3.4 The Original *TopSPA*

The *TopSPA* is an adaption of SPA for topics. The *TopSPA* utilizes topics for (a) initial energy assignment, and (b) topic-specific energy dissipation factor for each given vertex for each given topic. Like the traditional SPA, *TopSPA* is an iterative process. The spreading activation process is carried out per topic. The universal set of topics is given. The process is carried out by learning each user's affinity to a given topic using initial observation and subsequently using this observation to predict the flow of topical information (energy). It constructs the dynamic dissipation factor, d_{v_i, t_l, p_k} , for a given user, for a given topic, at a given stage. Let α_{v_i} represent the inherent tendency of v_i to pass information irrespective of the topic. Let β represent the scale parameter. Let p_k be the vertex's total energy, for t_l , at a given time.

$$d_{v_i, t_l, p_k} = \alpha_{v_i} + (1 - \alpha_{v_i}) e^{1 - \beta \cdot (p_k | t_l)} \quad (4)$$

3.5 Modified *TopSPA*

Hate has multiple forms, as mentioned earlier. We hypothesise that instead of modelling the spread of hate as a single form, we should model the spread such that it factors in the hate forms present in the hateful content. Therefore, we decide to borrow the existing *TopSPA* model from Kuntal et al. [4] to model the spread of hateful tweets for each topic. We make the following modifications to the original *TopSPA* model to capture the spread of hateful forms effectively.

Dynamic Decay Factor Selection. The decay factor d is an essential factor in the SPA. The decay factor d determines how much energy a node allows to flow in the system. In *TopSPA*, d is chosen with the premise that a user would not repeatedly pass on the same topical information. Hence, repeated exposure to a topic results in less energy flow from a node. However, we hypothesize that hateful forms are different from regular topics. A hateful user for a topic keeps retweeting/quoting the hateful tweets belonging to the topic. Therefore, we keep d constant for a hateful form in our approach and experiment with multiple values of d to find the best in our setting.

Seed User Selection. Unlike *TopSPA*, our seed user selection considers the nature of the hate forms from the general public's perspective. We observe that there are certain hate forms to which users do not show strong affinity. These hateful forms are mainly around abusive words which is a common way to express hate, while there are hate forms, to which people show strong association such as hate against India and Pakistan. Therefore, we keep the affinity threshold for users classified as seed users, dynamic across the hateful forms for selecting seed users. This helps ensure the topics widely discussed, and not many users show extreme affinity, a suitable set of seed users is selected. For generic hate forms, the smaller threshold for hate affinity is chosen while for specific hate forms, a slightly larger threshold is selected.

4 Experiments and Results

4.1 Dataset and Preprocessing

Dataset. We use the dataset provided by [10]. This dataset contains 200 most recent tweets of 100,386 users, totalling around 19M tweets. It also contains a retweet induced graph of the users. It has 2,286,592 directed edges. The retweet-induced graph is a directed graph $G = (V, E)$ where each node $u \in V$ represents a user in Twitter, and each edge $(u_1, u_2) \in E$ represents a retweet in the network, where the user u_1 has retweeted user u_2 . Retweet graphs have been largely used in the social network analysis, with previous work suggesting that retweets are better than followers to judge the influence of users [2]. We convert the edge list to an adjacency matrix and normalise, as mentioned earlier. We use this normalised matrix for both SPA and *TopSPA*.

Furthermore, every tweet is categorised as an original tweet, retweet or quote (retweet with a comment). Out of the 100,386 users, labels (hateful or normal) are available for 4,972 users, out of which 544 users are labelled as hateful and the rest as normal. Though the original dataset does not have labels for the tweet content, we use it for two reasons, a) it is the most extensive collection of users marked as hateful and b) it allows us to analyse the retweet-induced graph structure.

4.2 Annotation

To show the efficacy of our ensemble model, we manually annotate the tweets as hateful or not. Annotating 19M tweets is a costly and time-consuming process. Hence, we annotate a subset of tweets. We pick 10 original tweets produced by each of the 544 users labelled as hateful in the dataset for two main reasons. Firstly, annotating the tweets of hateful users ensures that we get an adequate number of hateful tweets. Secondly, this annotated set serves as a validation ground truth for downstream tasks in our research. After preprocessing the tweets, 4,720 original tweets are left. The tweets were annotated by a group of four independent annotators whose primary language is English. The Inter-Annotator Agreement (IAA) score (Cohen κ) was 0.87. A high κ score was obtained due to a singular class, hateful or not.

4.3 Hate Classification Results

We use datasets from two different domains, i.e. from toxic comments and hateful tweets. Model A learns from the dataset of toxic comments, while model B learns from tweets with hate speech against immigrants and women. This ensures the ensemble hate classifier is trained with the diverse aspects of hate. Both the models consist of a logistic regression classifier with Naive Bayes log count ratio as features. We implement the models using the scikit-learn¹ library. Further, we split both the datasets into an 80:20 train-test split. We empirically tune the parameters for the model to maximize the F1 score.

Model A is borrowed from a kernel from Kaggle posted for the competition “Toxic Comment Classification Challenge”². The dataset contains 160K labelled (hateful or non-hateful) comments. We add tf-idf with unigrams and bigrams and set minimum document frequency³ to 3 and maximum document frequency⁴ to 0.9. The model achieves an accuracy of 95% with a F1 score of 0.8 on the test data. This model achieves an accuracy of 72% with a F1 score of 0.69 on our annotated dataset.

Model B follows the same principal as model A. However, we modify the feature space by not specifying the minimum and maximum document frequency.

¹ <https://scikit-learn.org/stable/>.

² <https://www.kaggle.com/jhoward/nb-svm-strong-linear-baseline>.

³ This implies removing the terms that appear in less than 3 documents.

⁴ This implies removing the terms which appear in more than 90% of the documents.

Model B is trained on a dataset from a Semeval challenge called “Multilingual detection of hate speech against immigrants and women in Twitter (hatEval)”⁵. This dataset consists of 9,000 labelled tweets. The model achieves a accuracy of 80% with an F1 score of 0.7 and accuracy of 78% with an F1 score of 0.73 on the test and our annotated data.

The final predictions are performed separately using both the models, and the mean confidence score is used to predict a class (hateful or non-hateful). The ensemble model achieves an accuracy of 82% with an F1 score of 0.75.

We then proceed to classify our whole corpus of $19M$ tweets using our ensemble model. From the classification results, we infer that out of $1.6M$ quotes, 12.4% are hateful, out of $7.23M$ retweets, 8.36% are hateful, and out of $10.7M$ original tweets, 7% are hateful. To classify if a user is hateful or not, we analyse hateful posts in the ground truth hateful users given in the dataset. We find that hateful users have at least 50% of their posts classified as hate. Therefore, we label a user as hateful who satisfies this condition.

4.4 Data Segmentation

The posts present in the dataset are spanning from January 1, 2017, to September 30, 2017. The obvious choice to split the data into two-time segments is to have equal time windows. We explore this, but we realize that the posts’ distribution and the users who have made the posts are very skewed. The first time segment has significantly less number of posts as well as users. Therefore, we first split the data at per month granularity and look at the distribution of the posts and the users. Figure 2 shows the distribution of the number of posts and the number of unique users per month. We realize that the first time segment should span from January to August, while the second time segment spans the remaining months. We work with this time segmentation for the experiments. We find 50,032 unique users in this first segment with 3,51,318 edges in the retweet induced subgraph. In the rest of this paper, the first time segment is denoted as T1 while the second time segment is denoted as T2.

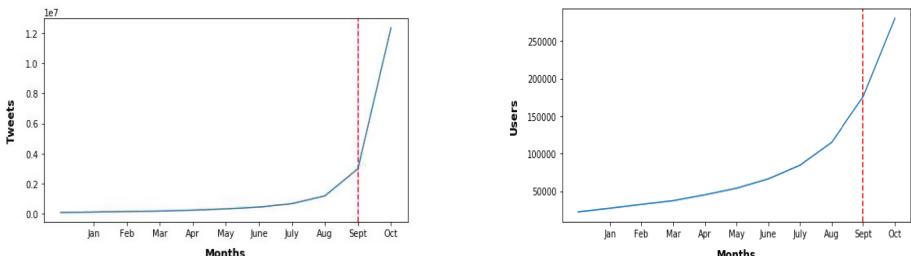


Fig. 2. Distribution of Tweets and Users with time

⁵ <https://competitions.codalab.org/competitions/19935>.

4.5 SPA Based Results for Hateful User Detection

We select all the hateful user in time segment T1 as the seed users for SPA. To be classified as hateful in a time segment, we only look at the posts made in that time segment. SPA related parameters are selected by experimentation. The Th_{SPA} is taken as 0.001. We vary the decay factor as 0.1, 0.3, 0.5 and 0.7. We let the SPA process run. We validate the approach based on the hateful users in time segment T2. After the SPA converges, each user receives some energy due to the flow of hate energy. A higher amount of energy corresponds to a higher likelihood of producing hateful content by the user. From the amount of energies each user accumulates in the SPA process, we construct a ranked list of users called SPA ranked list (RS), $RS = RS_1, RS_2, \dots, RS_m$, the higher the energy, the higher the rank of a user.

We also construct a ground truth ranked list of users in the time segment T2 using the posts classified as hateful. We compute the fraction of hateful posts made by each user. Higher the fraction of hateful posts by a user, higher is the rank. Let the ground truth ranked list be $RG = RG_1, RG_2, \dots, RG_m$. We use two metrics to quantify the SPA model's accuracy concerning the degree of overlap of the two ranked lists, precision and Jaccard coefficient (JC). For prediction set A and ground truth set B, we measure precision as $|A \cap B|/|B|$ while JC is computed as $|A \cap B|/|A \cup B|$. We vary d, the decay factor, as discussed earlier in the SPA process. We find that lower values of d produce better results for both precision and JC.

We compare results achieved for the best value of d with a baseline. The baseline finds the users who will produce hateful content are the direct neighbours of a hateful user, i.e. all the direct neighbours of hateful users shall start producing hateful content in the future. Table 1 shows accuracy of predicted hateful users using SPA against the baseline for $d = 0.1$. SPA outperforms the baselines for both the metrics, precision and Jaccard correlation coefficient, especially for lower overlap percentages.

Table 1. Result of SPA model for hateful users detection

Metric	Method	10%	20%	30%	40%
Precision	SPA (d = 0.1)	0.2628	0.3629	0.4424	0.52
	Baseline	0.1136	0.2211	0.3730	0.5441
JC	SPA (d = 0.1)	0.1512	0.2216	0.2840	0.3514
	Baseline	0.0602	0.1243	0.2311	0.3740

4.6 Topic Modelling

We create a document for each user by combining all the posts (tweets, retweets and quotes) classified as hateful by our hate classifier. We create a corpus of all the documents of the users. We explored LDA based topic modelling using

Mallet⁶ and Gensim⁷ as they use different sampling techniques. We looked at the coherence scores [11] and visualization of topics in terms of overlap using pyLDAvis⁸ while varying the number of topics from 7 to 10 as 7, 8 and 10. We find that Mallet produces more coherent and less overlapping topics. Therefore we use Mallet. Before coming up with a small number of topics, we experiment with large numbers such as 15 and 20. We observe that the produced topics have relatively less coherence and a high overlap as compared to a smaller number of topics. To select α and the number of topics, we perform a grid search where we vary alpha between 0.1 and 0.01 and the number of topics from 6 to 12. The number of iterations for each run is 500. We update the alpha and beta parameters every 10th iteration.

We observe that $\alpha = 0.01$ gives us a higher coherence score for the same number of topics as compared to $\alpha = 0.1$. From this, we infer that a document in our corpus contains just a few topics. We also compare the topic visualizations using pyLDAvis. We find that the best performing topic model, which has both a high coherence score and the least number of overlapping topics, is when $\alpha = 0.1$ and number of topics equal to 8. We use this topic model to compute a topic affinity vector for every hateful tweet in the corpus.

4.7 Top SPA Results

We repeat the SPA process for each latent topic present in the hateful tweets. Seed users for each hate form are selected from the time segment T1. Initially, we select seed users with a uniform threshold of 0.5 for the topic/hate form affinity for a user, but we realize that a uniform threshold does not perform well on a few topics. We discover that the number of seeds users in these topics is much less than the total users. We also find that these topics are generic in nature. Figure 3 shows that for the topics, 1, 6 and 8, the affinity does not vary much

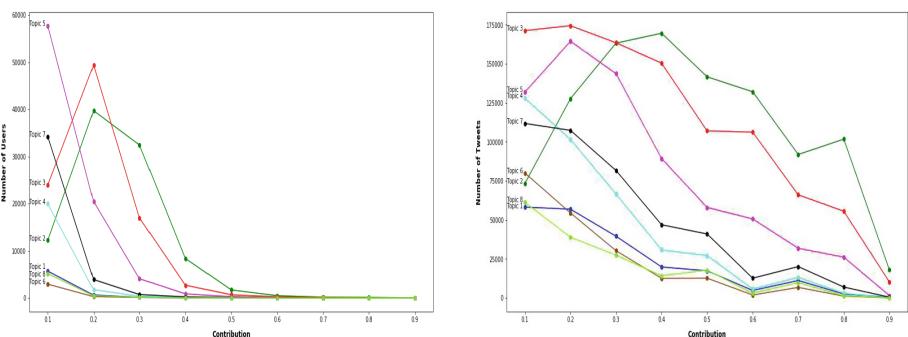


Fig. 3. Affinity of tweets with respect to the topics present

⁶ <http://mallet.cs.umass.edu/>.

⁷ <https://radimrehurek.com/gensim/>.

⁸ <https://pypi.org/project/pyLDAvis/>.

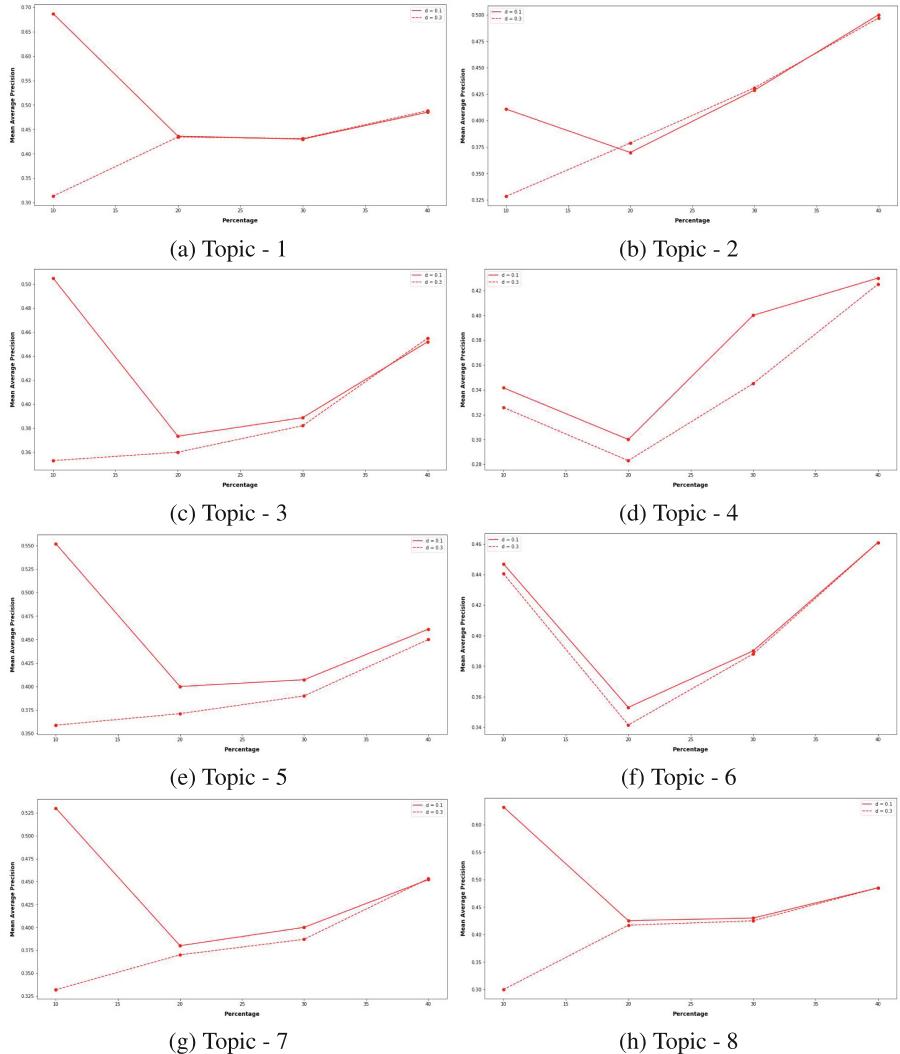


Fig. 4. Comparison of MAP for each topic after running our modified *TopSPA*

across the users. Further, many tweets show a strong association with topics 1, 6 and 8. Therefore, we decide to have a dynamic threshold for topic affinity, catering to each topic for seed user selection.

We compute dynamic affinity values for seed selection across multiple topics so that we get a reasonable number (at least 5% of total users) of seed users. We also vary the decay factor d as 0.1 and 0.3 for the dynamic affinity values. We compute the two metrics discussed earlier to show the efficacy of the model. Due to space constraint, we do not show the Jaccard Coefficient (JC) for every topic; rather we showcase the Mean Average Precision (MAP) results for each

topic and the mean of MAP across the topics. The results for each topic and the effect of varying d is shown in Fig. 4. We observe that our modified *TopSPA* also gives best results for lower values of d .

5 Conclusion

This paper addresses a very timely and relevant problem of hate spread modelling on social media platforms. We show that spreading and activation based methods are capable of capturing the spread of hate. We find that hateful content generated on social media platforms is not a blanket category but spans multiple forms. We propose to use latent topic modelling to detect the hate forms. We demonstrate on a publicly available Twitter dataset that modelling the spread of hate by factoring in its multiple forms gives us insights into their spread characteristics. Analyzing hate through its multiple forms also outperforms hate being treated as a blanket category.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *J. Mach. Learn. Res.* **3**(Jan), 993–1022 (2003)
2. Cha, M., Haddadi, H., Benevenuto, F., Gummadi, K.P.: Measuring user influence in Twitter: the million follower fallacy. In: Fourth International AAAI Conference on Weblogs and Social Media (2010)
3. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated hate speech detection and the problem of offensive language. In: Eleventh International AAAI Conference on Web and Social Media (2017)
4. Dey, K., Lamba, H., Nagar, S., Gupta, S., Kaushik, S.: Modeling topical information diffusion over microblog networks. In: International Conference on Complex Networks and their Applications, pp. 353–364. Springer (2018)
5. Golub, B., Jackson, M.O.: Naive learning in social networks and the wisdom of crowds. *Am. Econ. J.: Microecon.* **2**(1), 112–49 (2010)
6. Li, M., Wang, X., Gao, K., Zhang, S.: A survey on information diffusion in online social networks: Models and methods. *Information* **8**(4), 118 (2017)
7. Mathew, B., Dutt, R., Goyal, P., Mukherjee, A.: Spread of hate speech in online social media. In: Proceedings of the 10th ACM Conference on Web Science
8. Mathew, B., Kumar, N., Goyal, P., Mukherjee, A., et al.: Analyzing the hate and counter speech accounts on Twitter. arXiv preprint [arXiv:1812.02712](https://arxiv.org/abs/1812.02712) (2018)
9. Ratadiya, P., Mishra, D.: An attention ensemble based approach for multilabel profanity detection. In: 2019 International Conference on Data Mining Workshops (ICDMW), pp. 544–550 (2019)
10. Ribeiro, M.H., Calais, P.H., Santos, Y.A., Almeida, V.A., Meira Jr, W.: “Like sheep among wolves”: characterizing hateful users on Twitter. arXiv preprint [arXiv:1801.00317](https://arxiv.org/abs/1801.00317) (2017)
11. Röder, M., Both, A., Hinneburg, A.: Exploring the space of topic coherence measures. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pp. 399–408 (2015)

12. Romero, D.M., Meeder, B., Kleinberg, J.: Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on Twitter. In: Proceedings of the 20th International Conference on World Wide Web, pp. 695–704 (2011)
13. Wang, C., Yang, X.Y., Xu, K., Ma, J.: SEIR-based model for the information spreading over SNS. *Acta Electronica Sinica* **11**, 031 (2014)
14. Wang, Q., Lin, Z., Jin, Y., Cheng, S., Yang, T.: ESIS: emotion-based spreader-ignorant-Stifler model for information diffusion. *Knowl.-Based Syst.*
15. Wang, S., Manning, C.D.: Baselines and bigrams: simple, good sentiment and topic classification. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, pp. 90–94 (2012)
16. Xuejun, D.: Research on propagation model of public opinion topics based on SCIR in microblogging. *Comput. Eng. Appl.* **51**(8), 20–26 (2015)



Indirect Causal Influence of a Single Bot on Opinion Dynamics Through a Simple Recommendation Algorithm

Niccolo Pescetelli^{1,2(✉)}, Daniel Barkoczi², and Manuel Cebrian²

¹ New Jersey Institute of Technology, 323 Dr Martin Luther King Jr Blvd, Newark, NJ 07102, USA

niccolo.pescetelli@njit.edu

² Max Planck Institute for Human Development, 94 Lentzeallee, 14195 Berlin, Germany

Abstract. The ability of social and political bots to influence public opinion is often difficult to estimate. Recent studies found that hyper-partisan accounts often directly interact with already highly polarised users on Twitter and are unlikely to influence the general population’s average opinion. In this study, we suggest that social bots, trolls and zealots may influence people’s views not only via direct interactions (e.g. retweets, at-mentions and likes) but also via indirect causal pathways mediated by platforms’ content recommendation systems. Using a simple agent-based opinion-dynamics simulation, we isolate the effect of a single bot – representing only 1% of the population – on the average opinion of Bayesian agents when we remove all direct connections between the bot and human agents. We compare this experimental condition with an identical baseline condition where such a bot is absent. We used the same random seed in both simulations so that all other conditions remained identical. Results show that, even in the absence of direct connections, the mere presence of the bot is sufficient to shift the average population opinion. Furthermore, we observe that the presence of the bot significantly affects the opinion of almost all agents in the population. Overall, these findings offer a proof of concept that bots and hyperpartisan accounts can influence average population opinions not only by directly interacting with human accounts but also by shifting platforms’ recommendation engines’ internal representations.

Keywords: Bots · Opinion dynamics · Bayesian belief update · Recommender systems · Social influence

1 Introduction

This study uses an agent-based simulation to explore the interaction between bots and content recommendation algorithms. We test the hypothesis that when recommender systems mediate information access, bots can affect a population’s mean opinion not just by direct interactions with other nodes, but via skewing the training sample fed to the recommender system during training. Thus, by subtly affecting how a centralised recommender system represents a population’s preferences and patterns of content

engagement, a bot may influence content recommendation at the population level at a much larger scale. This indirect social influence may be more pervasive than direct influence because it would take place even in the absence of direct bot-human interaction.

The potential of algorithmic agents, commonly referred to as bots, to influence public opinion has been recently put under closer scrutiny. Early studies documented the potential effects of bots on skewing opinion distributions and polarizing opinions on social media users and voters [1–7]. These generalised concerns have mobilised platforms to improve algorithmic agents' automatic detection and removal [8–10]. On top of bot influence, influence networks online are characterised by several other phenomena acting together on public opinions, such as human trolls, fake accounts, pink-slime newspapers and “fake news” [11–15]. In this paper, we estimate algorithmic influence lower bound by focusing on the effect of a single algorithmic agent on a population. Our findings can theoretically be generalised to other ‘pre-programmed’ agents of media manipulation, such as partisan accounts and human trolls, that share similar features, such as “stubbornness”.

Recently, several researchers have attempted to measure the effect of hyper-partisan content by looking at social media data from the 2016 USA presidential election [16, 17]. These studies suggest that sharing and consuming fake or hyper-partisan content was relatively rare relative to the total volume of content consumed. One study in particular [18] attempted to measure the effect of being exposed to Russia’s Internet Research Agency (IRA) content on opinion. The authors found that interactions with highly partisan accounts were most common among respondents with already strong ideological alignment with those opinions. The researchers interpreted these findings as suggesting that hyper-partisan accounts might fail to significantly change opinions because they primarily interacted with those who were already highly polarised. This phenomenon is also known as “minimal effect” [19–21]. Furthermore, exposure to partisan ideology can counter-intuitively strengthen confidence in one’s own belief [22–24]. Overall, these findings cast doubt on the centrality and impact of bot activity on political mobilisations’ coverage. Notwithstanding the well-documented spread of bots and troll factories on social media, their effect on influencing opinions may be limited.

The studies reviewed above were primarily concerned with *direct* influence among agents. Although common in many offline and online settings, we argue that direct influence does not take into account the complexity of the digital influence landscape. Direct influence assumes that exposure to another person’s belief (e.g. an advisor) produces changes to a privately held belief [25–28]. However, this simple model of social influence may be outdated in the modern digital environment. Although direct interactions on most online platforms do occur (e.g. friends exchanging messages and users tweeting their views), information exchange is also mediated by algorithmic procedures that sort, rank, and disseminate or throttle information. The algorithmic ranking of information can affect exposure to specific views [29]. Recommender systems can learn population averages and trends, forming very accurate representations of individual and collective news consumption patterns [30, 31]. One crucial

difference between traditional social interactions and machine-mediated interactions is that in the latter case, single users can influence not only other people's beliefs but the "belief" of the content curation algorithm (i.e., its internal model). We call this indirect influence. Furthermore, as nodes are connected in a network, another form of indirect influence is via intermediary nodes. In other words, a bot may directly influence one human but indirectly influence all the humans that this human is connected to.

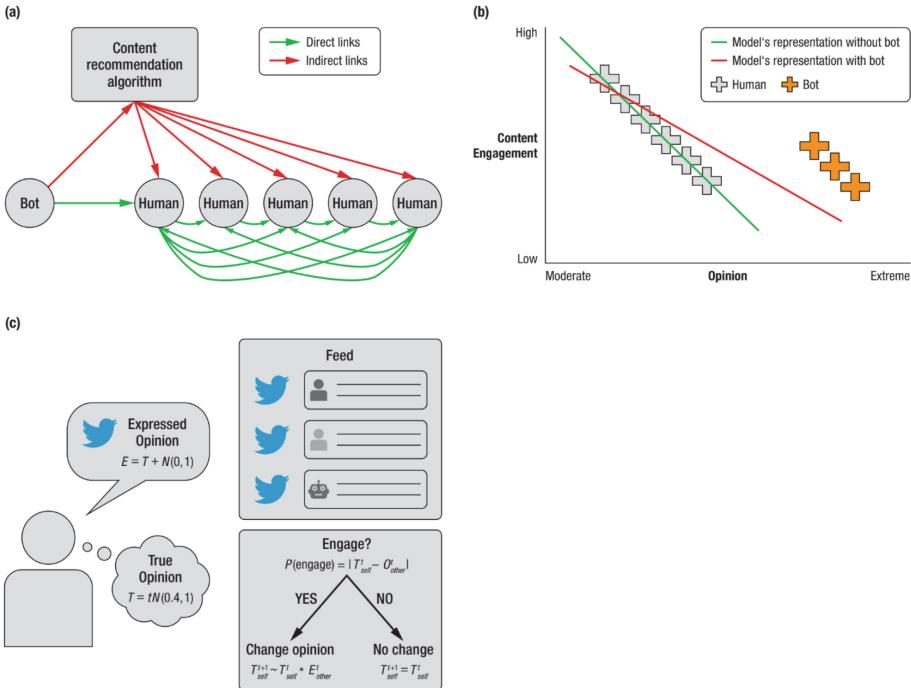


Fig. 1. The indirect influence of bots on social information networks. (a) Representation of opinion dynamics network mediated by a content recommender system (grey box). Bot and Human agents (circles) consume and share content on a platform. A bot agent can influence human opinions via direct (green links) interaction with human agents (e.g. retweets, mentions, likes etc.) or indirectly (red links) via affecting the inner representation of the content recommendation algorithm. (b) Schematic representation of the effect of bot presence on the internal representation learned by a simple recommender system trained to predict a user's engagement with content. The inclusion of the bot behaviour in the training set skews the model to think that engagement with extreme content is more likely than it would be without the bot presence. (c) agents in the simulation were modelled to include a private true opinion and an expressed public opinion. On every round, they were presented with the public opinion of their neighbours and decided whether to engage with this content or not, according to a pre-defined engagement function. Opinion change happened only if the agent engaged with the content.

2 Methods

2.1 Overview

Figure 1A shows a mix of both direct (green) and indirect (red) links. In our simulation, we only focused on indirect links. We investigate a previously unexplored indirect causal pathway connecting social bots and individuals via a simple recommendation algorithm (Fig. 1A). We test the hypothesis that pre-programmed agents, like bots and troll factories, can disproportionately influence the entire population by biasing the training sample of recommender algorithms predicting user engagement and user opinions (Fig. 1B). We created two identical fully connected networks of 100 agents. The two networks differed only in the presence (or absence) of one bot. We initialised the two simulations using the same random seed, which allowed us to directly test the counterfactual of introducing a single bot in the network while holding all other conditions constant. Crucially, while human agents were all connected, the bot could only interact with the other users via the recommendation algorithm (Fig. 1a). Our simulation differs from previous work on opinion dynamics in two important ways. First, contrary to previous studies [27, 28], we distinguish between internally held beliefs and externally observable behaviour. We assume that observable behaviour represents a noisy reading of true internal beliefs. This assumption captures the fact that people on several online platforms, such as fora and social media, can form beliefs and change opinions simply by consuming content and never post or share their own [32]. One does not need to tweet about climate change to have an opinion about it. Similarly, this distinction between internally held and publicly displayed beliefs allows us to train the recommender algorithm only with externally observable behaviour rather than making the unrealistic assumption that the algorithm has direct access to a user’s unobservable opinion. We call all such externally observable behaviours (e.g. tweets, likes and reactions) ‘engagement’. Thus, both the recommender algorithm and agents must infer other agents’ underlying opinions from engagement behaviours. Second, we use a Bayesian opinion updating rule [23, 33, 34]. The Bayesian update offers a natural way to consider not just the opinion direction on a binary issue but also belief’s conviction and resistance to changes of mind or new information. This belief update rule produces non-linear dynamics that have been shown to reflect belief updates in laboratory experiments [23, 35]. Such non-linear dynamics reflect the fact that people who agree tend to reinforce each other’s beliefs and move to more extreme positions. In comparison, people who disagree tend to converge to more uncertain positions (although see [22]).

We simulate a simplified social network model where a recommender system learns and presents a personalised content feed to agents in the network. This feed contains the expressed opinions of other agents in the network. Each agent can observe and decide to interact with other agents’ opinions by updating and expressing their own opinions. In two separate but otherwise identical conditions, we manipulate whether a single bot is also part of the potential pool of agents that the recommender system draws upon to create the feeds. We study whether this bot can infiltrate the feed created by the recommender system by influencing the statistical relationships it learns.

Across a series of simulations, we quantify the effect of adding a single bot to a network of fully connected agents. We show that the bot can influence human agents even though no direct link exists between human agents and the bot. We conclude that in an information system where trained models control who sees what, bots and hyperpartisan agents can influence the whole user population by influencing the internal representation learned by the recommender algorithm. In other words, what the recommender system believes is as crucial as people's beliefs when studying network opinion dynamics. We discuss these findings in light of the contemporary debate on social media regulation.

2.2 Simulation Procedure

Agents. We simulate $N = 100$ agents connected through a fully connected network. Each agent i is represented by a true private *opinion* in the range $[0, 1]$ drawn from a truncated Normal distribution

$$T_i = tN(0.4, 1) \quad (1)$$

and by an *expressed opinion* representing a noisy observation of their true opinions:

$$E_i = T_i + N(0, 1) \quad (2)$$

On each time step, agents go through a two-step process:

Engagement. First, they decide whether or not to *engage* with content in their feed (see below). Content is the expressed opinions of other agents j ranked by the recommender system for each agent individually. Agents decide whether to engage with the content based on an engagement function defined as

$$P(\text{engage}_j) = \left| T_i^{t-1} - O_j^{t-1} \right| \quad (3)$$

Where O stands for *observed* and is the expressed opinion E of another agent j . We represent engagement as a binary decision. This engagement function makes it more likely that agents engage with content that is more distant from their own opinions, representing the tendency that people have online to engage with sensationalist or counter-intuitive content more than moderate content [15, 36]. We explore in Supplementary Material two different engagement functions: in the former, agents are more likely to engage with content close to their own opinion (*homophilous engagement*), Fig. S1. In the latter, they are equally likely to engage with similar or dissimilar content (*bimodal engagement*), Fig. S2.

Opinion Update. On every time step t when agents decide to *engage*, they update their own opinions using a Bayesian opinion update function:

$$T_i^t = \frac{T_i^{t-1} * O_j^{t-1}}{(T_i^{t-1} * O_j^{t-1}) + (1 - T_i^{t-1})(1 - O_j^{t-1})} \quad (4)$$

On timesteps when the agents decide not to engage, they keep their opinion from the previous timestep, time $t - 1$: $T_i^t = T_i^{t-1}$.

Feed. Each agent i is presented with a feed consisting of the *expressed opinions* of n other agents in the social network. This feed is created by a simple recommender system separately for each agent. The goal of the feed is to provide content that agents are likely to engage with (see *Engagement* above). To achieve this, we train a simple logistic regression using agents' binary engagement history as a dependent variable and the absolute difference between the agent's public opinion at time $t - 1$ and the opinion they observed in their feed as the independent variable. In other words, the model aims to learn the agents' engagement function by observing their prior engagement history and the content they observed in their feeds. To provide sufficient training data for the recommender system, we start the first 10 timesteps of the simulation by presenting agents randomly in the feeds. Notice that although we simulate a fully connected network, it is the feed that determines whether an agent will see or not see the content created by another agent.

Bot. The bot is represented as an agent that does not change its opinion but sticks to the same opinion throughout the simulation [4, 37, 38]. In separate conditions, we manipulate the degree to which this opinion is extreme (i.e. the distance from the mean opinion of the agents). We initialize the simulation with the following parameters: Mean agent opinion: $N(0.4, 1)$ and bot opinion = 0.8. This represents a situation where agents are not polarized or extreme, and the mean difference between agents and the bot is not very extreme. On each timestep (starting from $t = 10$ onwards), agents are presented with a unique feed based on which they decide whether to engage and update their opinions. Once each agent has made its decision, the simulation proceeds to the next timestep. We repeat the procedure for $t = 100$ timesteps and $r = 100$ replications. We record the opinions of each agent on each timestep, as well as the cases where the bot gets recommended to an agent. We simulate two conditions, one where the bot is present and one where it is absent. We initialize both simulation conditions with the same random seeds, thereby, producing completely identical simulation conditions except for the presence of the bot. This allows for precise measurements regarding the influence of the bot on the network.

3 Results

3.1 Population-Level Influence of the Bot on the Average Opinion

We start by looking at the population-level influence of the bot on agents' opinions. We define influence as situations where an agent is presented with bot content in its feed, decides to engage based on the content observed and thus changes its initial opinion. Figure 2a shows the mean opinion in the entire group over time for the two conditions (bot vs no bot). Note that for the first $t = 10$ timesteps, there is no change in opinion

since those trials serve as training samples for the recommender system, and therefore, presents agents in the feed randomly. From $t = 10$ onwards, we see a significant difference between the two conditions, with the bot shifting the average opinion of the population by 5% on average. This effect is also reflected by the average engagement levels in the population, as depicted by Fig. 2b. This effect holds across different initial opinion distributions and different bot opinions (Supplementary Materials). From $t = 10$ onwards, we observe a large jump in engagements, showing that the recommender system is becoming increasingly efficient at recommending content that agents will engage with. The presence of a bot leads to remarkably higher engagement levels, indicating that by getting recommended, agents are more likely to engage and shift their opinions as a result of interacting with the bot directly or indirectly.

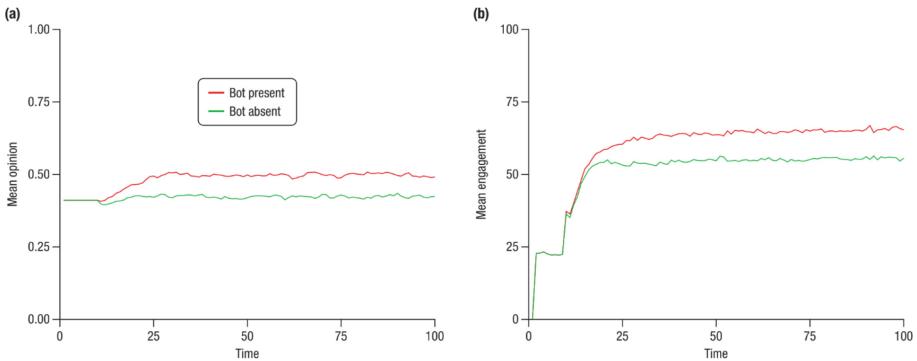


Fig. 2. Mean opinion and mean engagement in networks with and without bot influence. (a) Mean opinion of the agents over time. (b) Mean number of agents engaging in each timestep. Red: Condition where the bot is part of the social network, Blue: Condition where the bot is not part of the social network. A single bot can produce substantial changes in the mean opinion and mean engagement levels in the network.

3.2 Magnitude of Direct Bot Influence on Individual Agents

Here, we investigate the reasons underlying average opinion shifts more directly. Figure 3a shows the number of agents directly influenced by the bot on each timestep. By direct influence, we mean that the bot's content was recommended to an agent via the feed, and the agent decided to engage with the bot's content (and thus updates its private opinion based on the bot's content). On average, 2.5 agents engage with and change their opinions after observing the bot on any timestep, with an average opinion change of 30% (Fig. 3b). The spike observed in both graphs on the left-hand side is due to the fact that bots' opinions are less extreme and thus less engaging the more the population's opinion shifts towards the bot's opinion. The finding of low engagement and opinion shift replicates “minimal effect” findings online [18, 19]. Our finding only captures the direct influence from bot to agent but does not measure the bot's indirect influence by influencing an agent that will influence further agents. Our intuition is that indirect influence may be more pervasive and more pronounced, especially in online contexts where recommender systems facilitate information spread.

To measure this indirect n-th order influence of bots on agents we compare, in the next paragraph, the two simulation conditions (bot vs no bot) while using the same random seed and holding all other conditions constant.

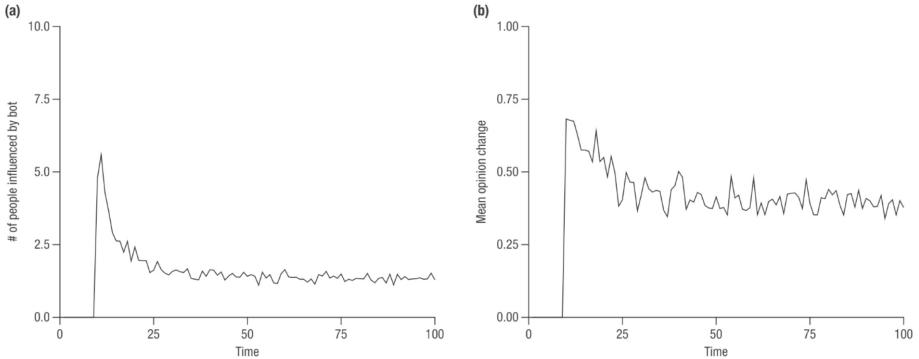


Fig. 3. Direct bot influence. (a) Average number of people influenced by the bot on each timestep. Influence is defined as when an agent is presented with content produced by the bot, engages with this content and shifts its own opinion. (b) Mean change in opinion for agents influenced by the bot on each timestep. A single bot can influence multiple people on each timestep and can produce substantial opinion change.

3.3 Individual-Level Shift in Opinion as a Result of Direct and Indirect Bot Influence

Figure 4a shows the difference in opinion between the same agent across the two simulation conditions, holding all other aspects of the simulation constant. Initialising the two simulations with identical parameters and random seed allowed us to isolate the effect of the bot. Estimating the within-agents effect improves our estimation of the effect of bot presence. Differences between the two counterfactual worlds reflect direct bot influence and all secondary effects caused by introducing the bot. Even though a small minority of agents was directly influenced by the bot (Fig. 3), we found that, compared to a counterfactual simulation, the bot had a pervasive indirect effect on the entire population (Fig. 4a). The magnitude of influence on opinion varied considerably, from 33 to 48% points, but virtually all agents showed shifted opinions compared to the control condition. This effect is explained by agents observing other agents that might have interacted with the bot, leading to a trickle-down effect of the bot's opinion on other agents who might not have interacted with the bot at all. Our model shows that bots' influence is magnified when we account for indirect influence, either via the recommender system or via other intermediary agents. This is a striking result that indicates that a single bot can have a much stronger and lasting effect beyond individuals it directly interacts with. This finding seems to suggest that studies focusing only on direct influence (bots' influence on people they directly interacted with) might be an underestimation of the actual capacity of a bot to bias and sway a population's opinion dynamics.

Finally, the above results assumed that the average opinion in the population is N (0.4,1) and the bot opinion is 0.7. Thus, the results are specific to this parametrization of our model. To test the generalisability of our conclusion, we explore the sensitivity of our results to different values of agent and bot opinion. Figure 4b shows a heatmap where the x-axis shows different values of the bot opinion and the y-axis shows the mean opinion in the population. The results remain qualitatively similar to those presented in the main text, with the bot having a stronger effect on the population when its opinion is more distant from the average opinion of the population. The results provide further support to the conclusion that a bot (here representing 1% of the total population) can have a disproportionate effect on population-level dynamics when we take into account indirect influence.

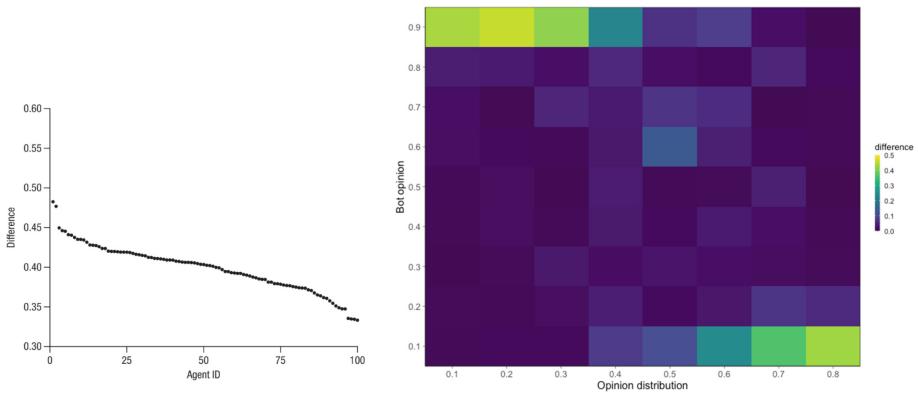


Fig. 4. Within-agents bot effect across the two simulations. (a) Difference between each agent’s opinion at time $t = 100$ between the two simulation conditions (bot vs no bot). This measures the total impact the bot has on the opinions of the same agents in the network. (b) Sensitivity of our results to different values of agent and bot opinion. A heatmap where the x-axis shows different values of bot’s opinion and the y-axis shows the mean opinion in the population. The results remain qualitatively similar to those presented in the main text, with the bot having a stronger effect on the population when its opinion is more distant from the average opinion of the population.

4 Discussion

This paper investigated the indirect influence that programmed agents, such as bots, trolls and zealots, can have via recommender systems. We find that a single bot can substantially shift the mean opinion and mean engagement of the population compared to a control condition without a bot. Even though only a minority of ‘human’ agents (2.5%) directly engaged with the bot’s content, the bot disproportionately affected the average shift in opinion observed in the population (30%). Notably, virtually all agents in the population were influenced by the bot presence, with opinion shifts ranging from 2 to 48% points.

This latter result would be unlikely if bots could influence human agents only via direct exposure. As bots represent only a minority of the population of agents (1% in

our simulation), it is unlikely that they can interact with and directly influence all other agents. Our findings show that a simple recommender system (a logistic regression in our simulation) dramatically increases the influence of a bot on the population. Thus, our first contribution is in advancing the debate around bots' influence and media manipulation. They highlight a previously unexplored phenomenon and draw attention to a subtle yet potentially pervasive phenomenon. Contrary to previous studies investigating social media bots, our work does not model direct interactions between bots and human agents (arguably representing a minority of interactions) but focuses on indirect effects via recommendation systems. Our findings highlight that malicious agents, such as bots and trolls factories, can massively increase their influence by infiltrating the internal representations of trained models tasked with content filtering. Our second contribution is that our setup allows us to compare counterfactual worlds, thus directly strengthening causal inference. We initialised both control (without-bot) and treatment (with-bot) simulations with the same parameters and random seed. Furthermore, effects on opinion shifts and engagement were calculated at the individual node level, thus measuring the effect of our treatment (bot presence) on the opinion dynamics and engagement of virtually identical human agents.

Although it may be difficult to manipulate these systems outside the lab, researchers have recently successfully inferred the hidden mechanisms underlying several proprietary algorithms by systematically prompting them [39–41]. We acknowledge that our findings are specific to our parametrization of the simulation used in this study. Although we characterized part of this parameter space in Fig. 4b, more work needs to be done, e.g. exploring different noise distributions and opinion update functions, and measuring standard errors across runs. Future studies should also investigate the effects of network size and alternative network structures on bot influence. Finally, our study used a simple logistic model to predict engagement. Although real recommender systems are much more complex machines, the effects highlighted in our findings are likely to affect, at least to some degree, any content filtering algorithm that extrapolates the behaviour of one user to another. This is the case in most recommender systems. More complex recommendation systems, such as collaborative filtering, may still be affected by the same dynamics highlighted here as long as they use population averages to predict individual preferences [31, 42–45].

The last contribution of this work is the use of a Bayesian opinion update model that captures dynamics of belief conviction, uncertainty and probabilistic judgments [23, 33, 34]. We selected this opinion update model as it offers several advantages over more common linear opinion-dynamics models [27, 28]. First, it can be seen as a normative rational model of opinion update. This feature allows us to quantify a best-case scenario, namely, the impact of bots if people were rational. Second, it naturally represents confidence as the distance from the maximum uncertainty point (50%). Third, encounters with agreeing agents tend to increase one's belief conviction, while encounters with disagreeing agents increase uncertainty. While linear updates may better model estimation tasks, Bayesian updates may better represent belief convictions and partisan affiliations, i.e. cases where interaction with like-minded individuals makes you more extreme.

We conclude by suggesting possible ways to reduce the risk of public opinion manipulation online. Although improving the detection and removal of automated

accounts remains an important strategy, a more valuable approach might be to better regulate recommender and filtering algorithms to make them more transparent and auditable, and thus resilient to manipulation.

Supplementary Materials

Different Engagement Functions In the main text, we presented results assuming that agents are more likely to engage when the distance between their own opinion and the other agent's opinion is high. Here we study the sensitivity of our results to other engagement functions. Figure S1 shows the same results as Fig. 2 in the main text. Instead of assuming that agents are more likely to engage when content is dissimilar, we assume that agents are more likely to engage when the observed content is similar. In Fig. S2, we study a bimodal engagement function where agents are more likely to engage with very similar or very dissimilar content and less likely to engage with content that is neither too similar nor too dissimilar.

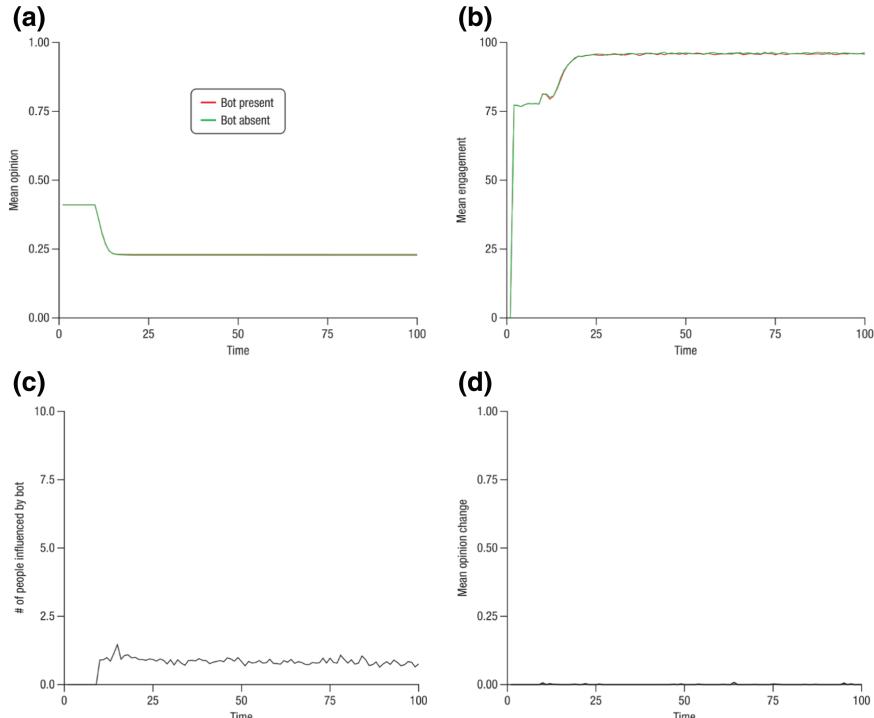


Fig. S1. Homophilous engagement function. Agents are more likely to engage with content in their feed that is closer to their own opinions. (a) population's mean opinion; (b) population's mean engagement; (c) the number of people influenced by the bot. (d) agents' mean opinion shift. The analysis shows that the results reported in the main text might be sensitive to the specific engagement function used by the agents to choose which content items they engage with.

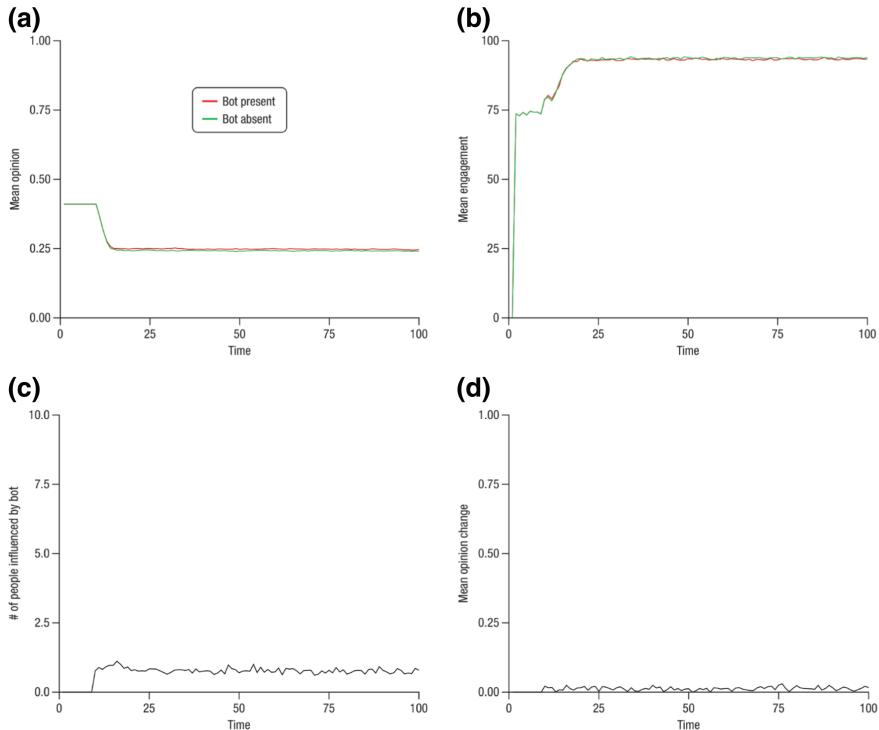


Fig. S2. Bimodal engagement function. Agents' engagement with content follows a binomial distribution with bimodal probability for content that is close to the target agent's private opinion as well as content that is distant from the agent's opinion. Content that falls between these two extremes is less likely to generate engagement. (a-d) population's mean opinion, population's mean engagement, number of people influenced by the bot and mean opinion change, as a function of time. Notice that contrary to the main text results (Fig. S2) here no difference emerges between conditions. This is likely due to the bimodal engagement function. Agents using this engagement function were more likely to engage with content that was similar to their own opinion. According to the Bayesian update rule (Eq. 4), this preference for similar content generates escalation dynamics that lead to agents reinforcing their own opinion [23] and thus being deaf to the bot's different opinion.

References

1. Bessi, A., Ferrara, E.: Social bots distort the 2016 US Presidential Election online discussion. *SSRN* **21**, 14 (2016)
2. Lerman, K., Yan, X., Wu, X.-Z.: The ‘Majority Illusion’ in social networks. *PLoS ONE* **11**, e0147617 (2016)
3. Broniatowski, D.A., et al.: Weaponized health communication: Twitter bots and Russian trolls amplify the vaccine debate. *Am. J. Public Health* **108**, 1378–1384 (2018)
4. Stewart, A.J., et al.: Information gerrymandering and undemocratic decisions. *Nature* **573**, 117–121 (2019)

5. Paul, C., Matthews, M.: The Russian ‘firehose of falsehood’ propaganda model. Rand Corporation 2–7 (2016)
6. Shao, C., et al.: The spread of low-credibility content by social bots. *Nat. Commun.* **9**, 4787 (2018)
7. Stella, M., Ferrara, E.D., Domenico, M.: Bots increase exposure to negative and inflammatory content in online social systems. *Proc. Natl. Acad. Sci. U.S.A.* **115**, 12435–12440 (2018)
8. Howard, P.: How political campaigns weaponize social media bots. *IEEE Spectrum* (2018)
9. Ferrara, E., Varol, O., Davis, C., Menczer, F., Flammini, A.: The rise of social bots. *Commun. ACM* **59**, 96–104 (2016)
10. Ledford, H.: Social scientists battle bots to glean insights from online chatter. *Nature* **578**, 17 (2020)
11. Hurtado, S., Ray, P., Marculescu, R.: Bot detection in Reddit political discussion. In: *Proceedings of the Fourth International Workshop on Social Sensing*, pp. 30–35. Association for Computing Machinery (2019)
12. Linville, D.L., Warren, P.L.: Troll factories: the internet research agency and state-sponsored agenda building. *Resource Centre on Media Freedom in Europe* (2018)
13. Aral, S., Eckles, D.: Protecting elections from social media manipulation. *Science* **365**, 858–861 (2019)
14. Tucker, J.A., et al.: Social Media, Political Polarization, and Political Disinformation: A Review of the Scientific Literature (2018). <https://doi.org/10.2139/ssrn.3144139>
15. Vosoughi, S., Roy, D., Aral, S.: The spread of true and false news online. *Science* **359**, 1146–1151 (2018)
16. Guess, A., Nagler, J., Tucker, J.: Less than you think: prevalence and predictors of fake news dissemination on Facebook. *Sci. Adv.* **5**, eaau4586 (2019)
17. Allen, J., Howland, B., Mobius, M., Rothschild, D., Watts, D.J.: Evaluating the fake news problem at the scale of the information ecosystem. *Sci. Adv.* **6**, eaay3539 (2020)
18. Bail, C.A., et al.: Assessing the Russian Internet Research Agency’s impact on the political attitudes and behaviors of American Twitter users in late 2017. *Proc. Natl. Acad. Sci.* **117**, 243–250 (2020)
19. Zaller, J.R.: *The Nature and Origins of Mass Opinion*. Cambridge University Press, Cambridge (1992)
20. Endres, K., Panagopoulos, C.: Cross-pressure and voting behavior: evidence from randomized experiments. *J. Polit.* **81**, 1090–1095 (2019)
21. Kalla, J.L., Broockman, D.E.: The Minimal persuasive effects of campaign contact in general elections: evidence from 49 field experiments. *Am. Polit. Sci. Rev.* **112**, 148–166 (2018)
22. Bail, C.A., et al.: Exposure to opposing views on social media can increase political polarization. *Proc. Natl. Acad. Sci.* **115**, 9216–9221 (2018)
23. Pescetelli, N., Yeung, N.: The effects of recursive communication dynamics on belief updating. *Proc. Roy. Soc. B: Biol. Sci.* **287**, 20200025 (2020)
24. González-Bailón, S.D., Domenico, M.: Bots are less central than verified accounts during contentious political events. *Proc. Natl. Acad. Sci. U.S.A.* **118**, 1–8 (2021)
25. Flache, A., et al.: Models of social influence: towards the next frontiers. *J. Artif. Soc. Soc. Simul.* **20**, 1–31 (2017)
26. Deffuant, G., Neau, D., Amblard, F., Weisbuch, G.: Mixing beliefs among interacting agents. *Adv. Complex Syst.* **03**, 87–98 (2000)
27. DeGroot, M.H.: Reaching a consensus. *J. Am. Stat. Assoc.* **69**, 118 (1974)
28. Friedkin, N.E., Johnsen, E.C.: Social influence and opinions. *J. Math. Sociol.* **15**, 193–206 (1990)

29. Bakshy, E., Messing, S., Adamic, L.A.: Exposure to ideologically diverse news and opinion on Facebook. *Science* **348**, 1–4 (2015)
30. Das, A., Datar, M., Garg, A., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: Proceedings of the 16th International Conference on World Wide Web, pp. 271–280 (2007)
31. Pipergias Analytis, P., Barkoczi, D., Lorenz-Spreen, P., Herzog, S.: The structure of social influence in recommender networks. In: Proceedings of The Web Conference 2020, pp. 2655–2661. Association for Computing Machinery (2020)
32. Lazer, D.: Studying human attention on the Internet. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 21–22 (2020)
33. Pescetelli, N., Yeung, N.: The role of decision confidence in advice-taking and trust formation. *J. Exp. Psychol. Gen.* (2020). <https://doi.org/10.1037/xge0000960>
34. Harris, A.J.L., Hahn, U., Madsen, J.K., Hsu, A.S.: The appeal to expert opinion: quantitative support for a Bayesian network approach. *Cogn. Sci.* **40**, 1496–1533 (2016)
35. Pescetelli, N., Rees, G., Bahrami, B.: The perceptual and social components of metacognition. *J. Exp. Psychol. Gen.* **145**, 949–965 (2016)
36. Lazer, D.M.J., et al.: The science of fake news. *Science* **359**, 1094–1096 (2018)
37. Karan, N., Salimi, F., Chakraborty, S.: Effect of zealots on the opinion dynamics of rational agents with bounded confidence. *Acta Phys. Pol. B* **49**, 73 (2018)
38. Yildiz, E., Acemoglu, D., Ozdaglar, A.E., Saberi, A., Scaglione, A.: Discrete opinion dynamics with stubborn agents. *SSRN Electron. J.* <https://doi.org/10.2139/ssrn.1744113>
39. Ali, M., et al.: Discrimination through optimization: how Facebook’s ad delivery can lead to biased outcomes. *Proc. ACM Hum.-Comput. Interact.* **3**, 1–30 (2019)
40. Hannak, A., et al.: Measuring personalization of web search. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 527–538. Association for Computing Machinery (2013)
41. Robertson, R.E., Lazer, D., Wilson, C.: Auditing the personalization and composition of politically-related search engine results pages. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW 2018, pp. 955–965. ACM Press (2018)
42. Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) *Recommender Systems Handbook*, pp. 1–35. Springer, Boston, MA (2011). https://doi.org/10.1007/978-0-387-85820-3_1
43. Das, A.S., Datar, M., Garg, A., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: Proceedings of the 16th International Conference on World Wide Web, pp. 271–280. Association for Computing Machinery (2007)
44. Koren, Y., Bell, R.: Advances in collaborative filtering. In: Ricci, F., Rokach, L., Shapira, B. (eds.) *Recommender Systems Handbook*, pp. 77–118. Springer, Boston (2015). https://doi.org/10.1007/978-1-4899-7637-6_3
45. Analytis, P.P., Barkoczi, D., Herzog, S.M.: Social learning strategies for matters of taste. *Nat. Hum. Behav.* **2**, 415–424 (2018)



Modelling the Effects of Self-learning and Social Influence on the Diversity of Knowledge

Tuan Pham^(✉)

The University of Chicago, Chicago, IL, USA
tuanhpham@pm.me

Abstract. This paper presents a computational model of acquiring new knowledge through self-learning (e.g. a Wikipedia “rabbit hole”) or social influence (e.g. recommendations through friends). This is set up in a bipartite network between a static social network (*agents*) and a static knowledge network (*topics*). For simplicity, the learning process is singly parameterized by α as the probability of self-learning, leaving $1 - \alpha$ as the socially-influenced discovery probability. Numerical simulations show a tradeoff of α on the diversity of knowledge when examined at the population level (e.g. number of distinct topics) and at the individual level (e.g. the average distance between topics for an agent), consistent across different intralayer configurations. In particular, higher values of α , or increased self-learning tendency, lead to higher population diversity and robustness. However, lower values of α , where learning/discovery is more easily influenced by social inputs, expand individual knowledge diversity more readily. These numerical results might provide some basic insights into how social influences can affect the diversity of human knowledge, especially in the age of information and social media.

Keywords: Topic diversity · Knowledge discovery · Bipartite network · Social influence · Self-learning

1 Introduction

As the world becomes more connected and the amount, as well as accessibility, of information in it increases, how do we learn an existing body of knowledge, while simultaneously updating with the new, incoming information? How diverse is knowledge acquired through social interactions? Coupled with limited cognitive capacity, do people become more specialized or generalized as a result, especially since specialization has implications for creativity and research productivity [9]?

Answering these questions might be difficult at this point without assessing simple cases of learning within static networks. Previous work addressed topic diversity analysis at the population level without considering the evolution of new topic acquisition [12], or used dynamic processes and analyses within only the intralayer networks [8]. A recent study discusses the involvement of social

interaction in innovation dynamics, but does not directly address the level of social influence nor analyze the resulting knowledge diversity in details [5]. Thus, I examined how different knowledge acquisition strategies could affect individual knowledge sets, as well as the diversity of knowledge for the whole population.

There are multiple ways a person could learn something new. This project focuses on only two ways: (1) active *self-learning* by acquiring new knowledge through related topics (Fig. 1b); and (2) through *social influence* as suggested by one’s social circle (Fig. 1c). An example of the former is following a “rabbit hole”, starting from an already-known topic on Wikipedia or a reference in a journal article’s bibliography section. On the other hand, examples of the second scenario include movie recommendations from friends or new research papers shared via social media. As humans tend to have limited capacity for learning, how would these two scenarios affect the diversity of knowledge of different *individuals* on average (specialists versus generalists), and of the entire social network *population* as a whole (e.g. can all topics be learnt)?

Considering only these two different ways of acquiring new topics in a probabilistic manner, I examine the diversity of knowledge, represented as different metrics based on the distribution of topics, as well as graph metrics. These are examined in simulations of randomly generated networks, with and without consideration of modularity within such networks. The results show that the self-learning process tends to improve topic diversity in the population manner, while recommendations through social influence would generally benefit individual diversity. Consideration of groups within the models have mixed effects at the individual level more so than the population level.

2 Methods

2.1 Model

General Description. All models considered here are binary undirected graphs. There are $n_a = 200$ agents and $n_t = 1000$ topics. Denote A and T as the symmetric binary adjacency matrices, respectively, of the agent graph G_a and topic graph G_t (Fig. 1a). The bipartite incidence matrix τ of size $n_t \times n_a$ represents the topics that the agents know about. It is assumed throughout that the intralayer edges are static while the interlayer edges could be *acquired* through an update process (described below). Once an interlayer edge is acquired, it is assumed to be persistent. At the initial stage, each agent is assigned at most $\tau_0 = 5$ topics with certain probabilities based on the models of the intralayer models (see below). There is also an upper limit topic capacity $\tau_{\max} = 50$ per agent, and the update process is only simulated until $1.2\tau_{\max} = 60$ time steps. Each parameter set (α , intralayer models, interlayer initialization) were simulated 5 times.

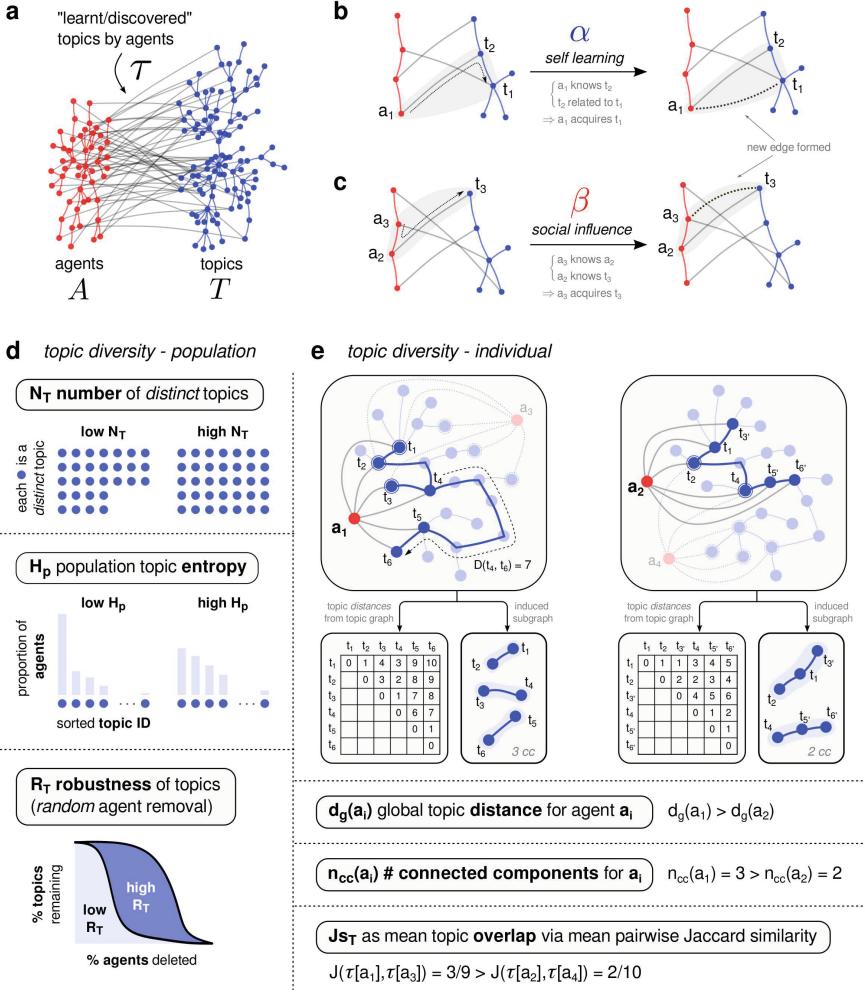


Fig. 1. Description of the topic update/discovery process in the model and the different knowledge diversity metrics. (a) Illustration of the intralayer agent graph (red) and topic graph (blue) with the interlayer edges (gray) representing the knowledge set of the agents. Gray triangles in (b) and (c) illustrate the update process either through learning/discovery by related topics (self-learning) or learning/discovery through friends (social influence). (d) Illustrations of different diversity metrics at the population level (each blue circle is a topic). (e) Illustrations of topic diversity metrics at the individual and local level (see Sect. 2.2 for detailed descriptions).

Update of Interlayer Edges. At each time step, at most one new topic is learnt per agent. The agent could acquire a new topic edge either through the self-learning strategy with α probability, by learning about the related topics of things an agent already knows about (Fig. 1b). On the other hand, with prob-

ability β (for simplicity assumed to be $= 1 - \alpha$), an agent could acquire a new topic edge by traversing its neighbors in the agent graph then to the topic graph (Fig. 1c). One way to implement this is below.

Define $\psi(X)$ as a column L1 normalization operation on a matrix X , in which each column vector \mathbf{x}_i of the matrix is normalized to $\mathbf{x}_i/\|\mathbf{x}_i\|_1$. Define the shorthand notation for the Heaviside function as $[x]_* = 1$ if $x > 0$, and 0 otherwise. At each time step, the probability matrix P (of same size as τ) with its column vector \mathbf{p}_i defines the probability agent a_i chooses a new topic. A way to define this probability is:

$$P = \alpha\psi\left(\left[\left[T\tau\right]_* - \tau\right]_*\right) + \beta\psi\left(\left[\left[\tau A\right]_* - \tau\right]_*\right) \quad (1)$$

$$\tau(t+1) \leftarrow \tau(t) + \text{sample}(P) \quad (2)$$

The multiplication steps perform the traversal through neighbors across the intralayer networks. The binarization and subtraction with the current τ simplifies the implementation, so that agents only learn new topics and avoid “being stuck” around too popular topics. Additionally, for simplicity I consider $\beta = 1 - \alpha$ so the process is only defined by α . Many other probabilities are ignored as well, for example serendipity (wandering or random discovery of new topics) and forgetting (removal or decrease of strength of interlayer edges).

Intralayer Random Models. For simplicity, the model types and model hyper-parameters (except only for the number of nodes) are the same for agent and topic graphs for each simulation.

Nonblock Models: The first approach is non-block networks. In the main text, only the scale-free (SF) network models are discussed, constructed by the linear preferential attachment models (PA) [1] (see Fig. 2). Additionally, other models are also analyzed: nonlinear PA models, Erdős–Rényi (ER) networks [3] with different connectivity probability, as well as small-world networks generated with the Watts–Strogatz (WS) models [11] (see Fig. 4a).

Block Models: Since there are usually communities in real-world networks, (researchers or papers within the same field), the stochastic block models (SBM) [4] are used to emulate this phenomenon with $k_a = k_t = 10$ groups for both agent and topic networks. A way to manipulate these models is to change the probability of connection within groups (p_{within}) or between groups (p_{between}). For simplicity, the former is kept fixed while varying the latter (see Fig. 3).

Interlayer Initialization. At the initialization stage, the probability of connection between a given agent and topic is uniform across topics. However, it is possible that other initialization strategies could influence the results. Hence, two different interlayer initialization strategies are introduced, one for *nonblock* intralayer models and one for *block* models. Whenever an initialization method is not mentioned, it is assumed to be the uniform random strategy.

For *nonblock* intralayer models, the probability of connecting to a certain topic could depend on its degree in G_t . A way to do this is to perform the $\text{softmax}(\{d_i\}; \beta_\sigma)$ on the degrees, basically transforming the degree sequence $\{d_i\}$ to a probability distribution. With $\beta_\sigma < 0$ (SOFTMAX_1), low degrees are favored; $\beta_\sigma = 0$ is equivalent to random initialization (SOFTMAX_2), while $\beta_\sigma > 0$ (SOFTMAX_3) favors high degree topics (Fig. 5a).

For *block* intralayer models, group correspondence could be used as a strategy for initialization as the number of groups are the same for both graphs. This could be parameterized by p_{sg} (Fig. 3) as the probability that agents and topics of the same group ID are connected. The chance $p_{sg} = \frac{1}{k_t} = 0.1$ would be equivalent to random initialization.

2.2 Diversity Metric

These diversity metrics are illustrated in Fig. 1d, e.

Population. Three population indices are defined, taken from an ecological perspective [10]. First, N_T is the number of distinct topics discovered when taking into account all agents' learnt topics, where higher values correspond to higher diversity. Second, H_p is the topic population entropy – the Shannon entropy from the discrete probability distribution of all the topics in the population (again, higher would suggest more diversity). Lastly, inspired by ecological network stability analysis [6], robustness can be calculated by cumulatively removing random agents and observing the remaining percentage of distinct topics. The area under this curve is the robustness R_T , higher values of which indicate that many agents need to be deleted to remove a large proportion of topics.

Individual. Three individual indices are calculated and the averaged computations across nodes (or pairs) of the agent graph are reported. First, d_g is the mean distance of the topics in each agent's learnt topics. In other words, if we define $D(t_i, t_j; G_t)$ as the shortest path distance in G_t between t_i and t_j , and an agent a_k 's topic set as $\tau[a_k] = \{t_h | \tau[t_h, a_k] = 1\}$ then $d_g(a_k) = \langle D(t_i, t_j; G_t) \rangle_{t_i, t_j \in \tau[a_k]}$. Higher values suggest that the agents know more outside of their comfort zone and tend more towards generalists. Another metric is the number of connected components $n_{cc}(a_k)$ in the induced subgraph $G_t(\tau[a_k])$, where higher values indicate there are many “islands” of topics that the agent knows about, again leaning towards generalist. Lastly, the mean pairwise Jaccard similarity J_{ST} between agents' topic sets are calculated, lower values suggest higher local diversity.

Group. With groups in the block intralayer models, I also calculated the entropy of the topic group distribution, in both the population sense H_{gp} and individual sense H_{gi} . More specifically, H_{gp} is the entropy of the 10 topic groups considering the group identities of all topics learnt by all agents. On the other hand, H_{gi} is the average entropy of each agent's individual topic entropy. These two quantities

are different; e.g. H_{gp} can be maximized (all groups uniformly distributed) while H_{gi} is 0 (each agent only learns about the topics within the same group).

2.3 Code Availability

The source code is at <https://github.com/tuanpham96/topic-diversity>. The simulations were run in parallel on Azure VM. Simulations, analyses and visualizations were performed in R, further illustrated in Inkscape.

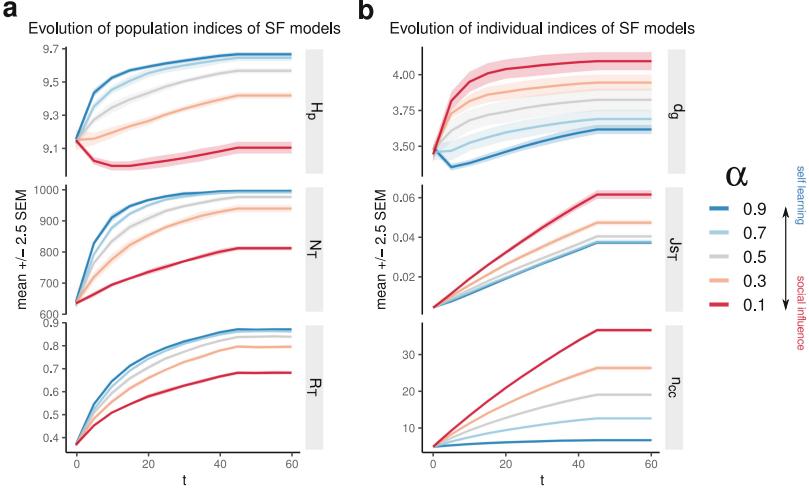


Fig. 2. Changes in population topic diversity indices (a) and individual diversity indices (b) of the scale-free (SF) intralayer models due to α (colors). H_p : topic population entropy; N_T : number of distinct topics; R_T : robustness to random removal of agents; d_g : mean distance of the subset of topics that agents know; J_{ST} : Jaccard similarity of topic set between agents; n_{cc} : number of connected components of induced subgraphs based on each agent's learnt topics. See Sect. 2.2 and Fig. 1d, e for more details. Each line plots the mean changes of 5 realizations for each index, analyzed every 5 steps.

3 Results

3.1 Nonblock Intralayer Models

The changes of the different diversity metrics for the scale-free networks are shown in Fig. 2 as an example to illustrate the tradeoff effect of self-learning versus social influence probability on population and individual diversity.

Generally, topic population diversity increases with self-learning probability α in terms of the topic entropy H_g and number of topics N_T . Through learning/discovery over time, low α could still achieve better population diversity. However, it does not seem likely for the worst case considered here, where

entropy does not even increase considerably past its initial value. The initial decrease of H_g when $\alpha = 0.1$ is because the agents start learning from each other, hence temporarily creating bias towards some topics, leading to decrease of entropy. It must be noted here that the entropies are already high initially due to initialization. However, taking the trends of both N_T and H_g into account, it is reasonable to say that higher α improves topic population diversity. Additionally, higher α leads to more robust retainment of the topics under random agent removal (i.e. higher R_T), though this is possibly partially an effect of higher N_T .

On the other hand, topic individual diversity usually decreases based on the chosen metrics. Increased α leads to decreased mean learnt topics distance d_g and number of components n_{cc} in the induced subgraphs. Intuitively, higher social influence – lower α – would allow the agents to access topics outside of their comfort zone more easily, hence their own subgraph of topics tend to be more generalized, whereas higher α leads to more specialization. Lastly, at the local level J_{ST} , lower α leads to more similarity between neighbors, hence lower local diversity. Although not analyzed, this hints at how social influence could create modularity in the learnt topic graph τ .

These trends are qualitatively consistent across different considerations of non-block models (Fig. 4), albeit some quantitative differences (especially in individual indices). Increases in α lead to higher topic population diversity (N_T, H_p), robustness (R_T) and local diversity (J_{ST}). On the other hand, such increases tend to result in loss of topic individual diversity (d_g, n_{cc}). Under consideration of degree-dependent initialization strategies (Fig. 5), favoring more obscure topics leads to the same trend as random initialization. However, initially favoring more popular topics is generally detrimental across different population, local and individual diversity indices, especially for those networks generated by preferential attachments (PA) models, possibly because learning more easily gets stuck in topics connected to the popular ones (last row in Fig. 5b).

In summary, in non-block intralayer models, higher self-learning (higher α) leads to higher topic diversity in a local and population context, but higher social influence (lower α) encourages individual topic diversity. Initializations that favor more popular topics have a negative effect on these different metrics.

3.2 Block Intralayer Models and Topic Group Diversity

As real-world networks usually contain communities within them, the stochastic block intralayer models (SBM) are used to model intralayer networks and investigate how diversity indices change due to α and network modularity. Generally, the trends for population diversity and robustness during the simulation are similar to those previously discussed (Fig. 6a). Such trends as a function of model modularity do not differ much. Looking at the group population entropy H_{gp} (Fig. 6b, bottom), only when the networks are less modular do those values show a difference, albeit very small.

In the individual perspective (Fig. 6c), I note that group modularity increases diversity indices d_g and n_{cc} , possibly because there are fewer long-range links. The trends for local diversity are roughly similar and not affected much by group

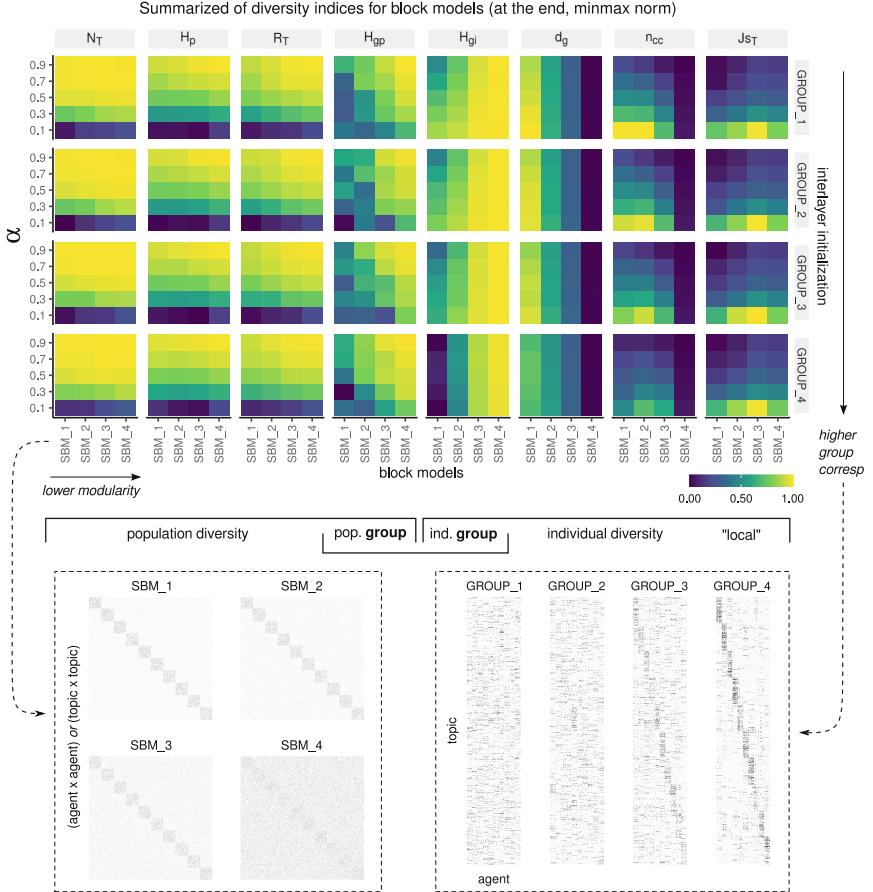


Fig. 3. Summary of population and individual diversity indices due to α , across different block models. Within each heatmap, the x-axis shows decreasing modularity of intralayer model (via increasing inter-modular connectivity), and the y-axis is α . The color represents values at the end of the simulations, and min-max normalized within each metric. From left to right are different diversity metrics. From top to bottom are different group correspondence initialization strategies.

modularity. Additionally, instead of only looking at topic group diversity in the population sense, one could also inspect it in the individual perspective. On average (Fig. 6b, top), for more modular intralayer networks, social influence benefits topic group diversity in the agents, because the agents would have more chances to learn out of their own comfort zone, especially if their initial topics belong to the same groups. With decreasing group modularity, these differences between α do not seem to matter any more.

The final values of these different metrics in Fig. 3, with different group-correspondence initialization strategies, reveal these effects more clearly.

More specifically, higher α and lower intralayer modularity generally leads to higher population topic diversity and robustness (N_T, H_p, R_T), whereas group correspondence initialization does not seem to have pronounced effects. Group modularity benefits individual diversity (d_g, n_{cc}) but high initial group correspondence would counter such effects, as learning gets stuck within communities. At the local level, group-correspondence does not seem to affect J_{ST} visibly. However, generally higher α and higher model modularity tends to decrease topic similarity, hence increasing local diversity.

For group entropies, low group modularity generally increases both topic group population (H_{gp}) and individual (H_{gi}) diversity. High initial correspondence seems to benefit group *population* diversity (although such benefits may be small, see Fig. 6b), but tends to decrease group *individual* diversity.

In summary, with consideration of intralayer block models, higher α (self-learning tendency) benefits more for population diversity and robustness, but less so for the group population diversity. In the presence of high social influence, high network modularity may hurt population diversity, regardless of initial group-correspondence. On the other hand, lower α is generally more beneficial for individual indices like those discussed with SF models, including group individual diversity, but either low network modularity or high group correspondence could be harmful for these metrics. At the local level, higher α and high network modularity generally decrease knowledge similarity between agents.

4 Discussion

In conclusion, with this simple toy model of topic discovery and a simple update rule that depends on the probability of traversing neighbors in bipartite networks, several interesting results emerge. First, increasing α (self-learning, traversing through interlayer edges first) leads to higher topic population diversity and robustness in several random models for the intralayer networks, including block and non-block models. However, such increase has drawbacks for topic individual diversity, as it reduces the chance for the agent nodes to acquire interlayer edges from topics that are usually distant from their comfort zone. Social influence, traversing through intralayer edges first (β route), would better influence individual diversity.

When intralayer groups are considered, group modularity may hurt the population diversity (some more than other) and more apparently the group individual entropy. Interestingly, it appears more beneficial for individual indices in term of graph distances and components. Although initial group correspondence does not have much effect on the population diversity, it has a dramatic drawback at the individual level (both entropy and graph metrics).

Limitations and Future Considerations. Though there are interesting results in a theoretical sense as a toy model, there are many limitations with the current study. One of the major drawbacks is the assumption that the knowledge space would eventually outgrow the agent population size hence considering

only $n_t > n_a$. Comparing the current estimate size of the indexed WWW¹ to the total internet users² might satisfy this assumption. However, a more realistic approach would be to consider the subset of the internet involved directly with knowledge, e.g. Wikipedia. Then the size of the topic network³ is around 2–3 orders of magnitude *smaller* than the potential agent network (See footnote 2), countering this assumption. Another assumption is that the intralayer models for the topic and agent graph are considered similar for simplicity and for practical reasons.

Future studies need to address these issues by integrating realistic knowledge and social network networks (or sampled versions of them) into the model, or at least the intralayer size ratios and generative models based on realistic network statistical properties. Other considerations include: different ratios between intralayer network sizes; inclusion of directed weighted edges (strengths could imply confidence in knowledge in τ); non-persistent interlayer edges; different update probabilities (serendipity, forgetting, mastering, ...); the cost of learning new subjects; delays in acquiring new knowledge; different versions of the update equation; the decreased disruptiveness in new knowledge discovery [7]; and, more importantly, the dynamic nature of the intralayer networks (e.g. [8]). Furthermore, future endeavours should also take into account performing the update process in real networks, which could be constructed using, as an example, the citation networks (agents as authors, papers as topics, groups as fields or subfields) or social networks (e.g. Twitter followers and hashtags) [12].

Additionally, further analyses should include examination of: the modularity changes in the bipartite τ [2] or in the projected graphs (for example, low α might start to create communities as evidenced by high Jaccard similarity in these simulations); the distribution of specialists and generalists; different local diversity definitions (e.g. topic entropy as a function of distance from a given agent); relationships between an agent's properties (e.g. degree, centrality) & position (e.g. hub, periphery) in G_a and the properties of their acquired topic sets (e.g. whether hub agents are more likely to also obtain peripheral arcane knowledge than peripheral agents); and lastly, persistent homology analyses (since the interlayer edges are defined as persistent here).

Acknowledgement. I would like to thank the reviewers of CNA 2021 for their comments, Dr. Mercedes Pascual and Dr. Sergio A. Alcalá Corona for their *Networks in Ecology and Evolution* course (University of Chicago), Dr. Julie S Haas (Lehigh University) and my friends Sam Nguyen, Poojya Ravishankar, Silas Busch for their discussion and feedback on the model, interpretations and writing. I would also like to acknowledge the Graduate Council Research & Personal Development Fund (University of Chicago).

¹ 30–50 billion as of Oct 3, 2021 (<https://worldwidewebsize.com/>).

² 4.66 billion internet users (<https://www.statista.com/statistics/617136/>).

³ 54.3 million total pages (6.4 million English) (https://wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia).

Supplementary Figures

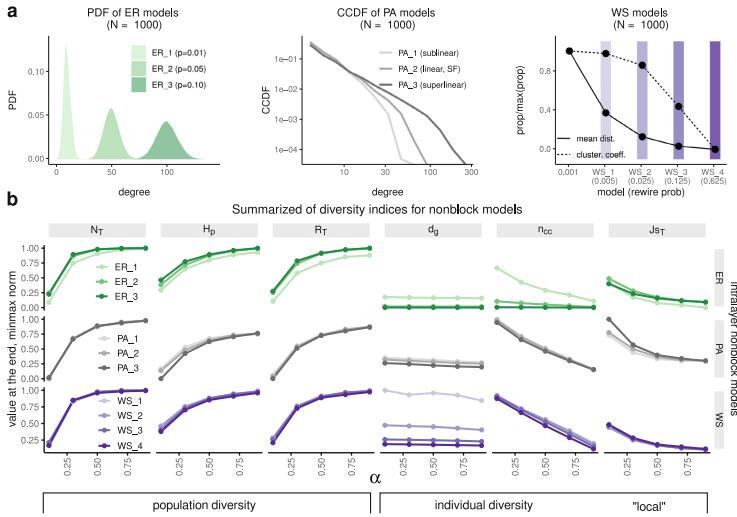


Fig. 4. Variations of nonblock intralayer models. (a) Set up of nonblock models. PA: preferential attachment, ER: Erdős–Rényi, WS: Watts–Strogatz (Sect. 2.1) (b) Changes of diversity indices for these models as a function α (Sect. 2.2)

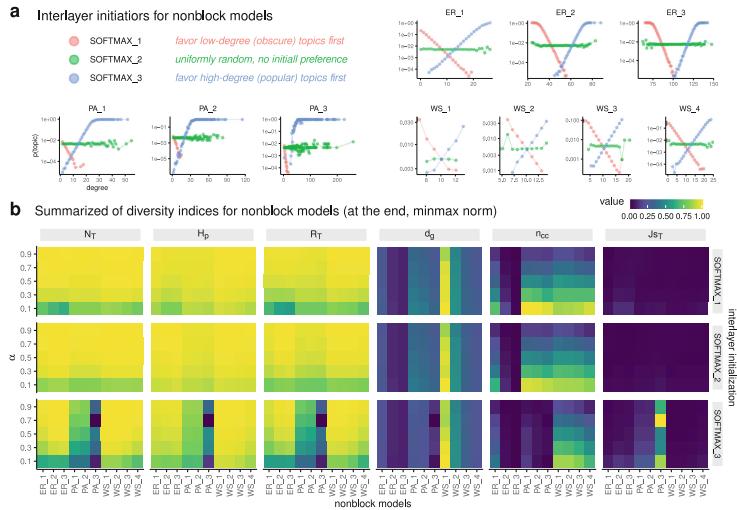


Fig. 5. Different initialization strategies for nonblock models (a) based on the topic intralayer degrees and (b) effects on population and individual diversity indices as a function of α . See Fig. 4a for names and illustrations of the different models.

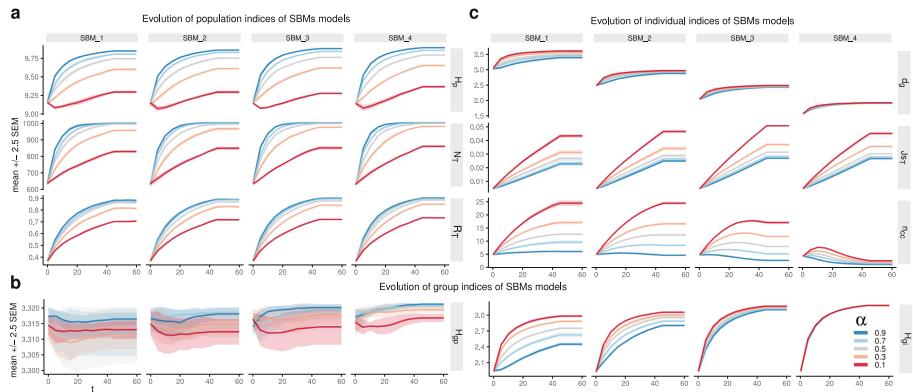


Fig. 6. Changes of population diversity indices (a), group diversity indices (b) and individual diversity indices (c) for the stochastic block intralayer models due to α .

References

1. Barabasi, A.L., Albert, R.: Emergence of scaling in random networks. *Science* **286**(5439), 509–512 (1999)
2. Dankulov, M.M., Melnik, R., Tadić, B.: The dynamics of meaningful social interactions and the emergence of collective knowledge. *Sci. Rep.* **5**, 12197 (2015)
3. Erdős, P., Rényi, A.: On random graphs I. *Publicationes Mathematicae* **6**, 290–297 (1959)
4. Faust, K., Wasserman, S.: Blockmodels: interpretation and evaluation. *Soc. Netw.* **14**(1), 5–61 (1992)
5. Iacopini, I., Di Bona, G., Ubaldi, E., Loreto, V., Latora, V.: Interacting discovery processes on complex networks. *Phys. Rev. Lett.* **125**(24), 248301 (2020)
6. Memmott, J., Waser, N.M., Price, M.V.: Tolerance of pollination networks to species extinctions. *Proc. Biol. Sci.* **271**(1557), 2605–2611 (2004)
7. Park, M., Leahey, E., Funk, R.: Dynamics of disruption in science and technology (2021)
8. Sun, Y., Latora, V.: The evolution of knowledge within and across fields in modern physics. *Sci. Rep.* **10**(1), 12097 (2020)
9. Teodoridis, F., Bikard, M., Vakili, K.: Creativity at the knowledge frontier: the impact of specialization in fast- and slow-paced domains. *Adm. Sci. Q.* **64**(4), 894–927 (2019)
10. Tuomisto, H.: A consistent terminology for quantifying species diversity? yes, it does exist. *Oecologia* **164**(4), 853–860 (2010)
11. Watts, D.J., Strogatz, S.H.: Collective dynamics of “small-world” networks. *Nature* **393**(6684), 440–442 (1998)
12. Weng, L., Menczer, F.: Topicality and impact in social media: diverse messages, focused messengers. *PLoS One* **10**(2), e0118410 (2015)



Activator-Inhibitor Model for Describing Interactions Between Fake News and Their Corrections

Masaki Aida¹(✉) and Ayako Hashizume²

¹ Tokyo Metropolitan University, Hino, Tokyo 191-0065, Japan
aida@tmu.ac.jp

² Hosei University, Machida, Tokyo 194-0298, Japan
hashiaya@hosei.ac.jp

Abstract. The problem of fake news continues to worsen in today's online social networks. Intuitively, it seems effective to send corrections as a countermeasure. However, there in actuality corrections can, ironically, strengthen attention to fake news, which worsens the situation. In this paper, we model the interaction between fake news and the corrections as a reaction-diffusion system and propose a framework that describes the mechanism that can make corrections increase attention to fake news. In this framework, the emergence of groups of users who believe in fake news is understood as a Turing pattern that appears in the activator-inhibitor model. Numerical calculations show that even if the network structure has no spatial bias, the interaction between fake news and the corrections creates a group that is strongly interested in discussing fake news.

Keywords: Online social network · Fake news · Reaction-diffusion system · Activator-inhibitor system · Turing pattern

1 Introduction

Social networking service (SNS) has become deeply widespread in our social life, and online user dynamics now greatly influence user behavior in both online society and the real world. In particular, fake news has had a great impact on the real world, and it is said that 70% of claims about Trump during the 2016 US presidential election were “false” or “mostly false” information [1]. Since it is impossible to eradicate fake news from the world, it is necessary to build a mechanism that is less susceptible to the adverse effects of fake news on the premise that modern society will have to coexist with fake news. For that purpose, it is necessary to understand the mechanism by which fake news spreads through online social networks (OSNs) and to establish an effective countermeasure.

The current belief is that proactively promoting the correct information that denies the content of fake news is an effective solution. Certainly, this seems to

be intuitively valid. However, in reality, the results are not so simple. A case has been reported in Japan where the information that corrected fake news did not work properly [2, 3].

The details of the matter are as follows. First, in late February 2020, at the beginning of the COVID-19 epidemic, a claim that toilet paper was out of stock was posted on SNS. The claim was that shortages of toilet paper were imminent as the production area of toilet paper, in China, was negatively impacted by COVID-19. In reality, there was sufficient inventory and production capacity in Japan, so stock shortages never occurred with normal demand. For this reason, this post was non-factual fake news. At this point, the fake news itself did not receive much attention and did not directly trigger the hoarding of toilet paper. After that, to counter this fake news, corrections were disseminated basically saying that toilet paper was not in short supply and that the original report was fake news. Many users who received this correction predicted that many other people would try to hoard toilet paper due to the fake report, which stimulated the sense of crisis. In addition, the spread of the corrections attracted the attention of users to the fake report, and as a result, the spread of fake news was actually promoted. As a result, the dissemination of the corrections to counter the fake report stimulated the hoarding of toilet paper, resulting in a shortage of toilet paper, ironically.

This case reinforced the difficulty of countering fake news in that the dissemination of correct information does not always lead to good results. In particular, since corrections can activate fake news, it is suggested that it is necessary to consider the interaction between fake news and the corrections in order to properly evaluate the effect of any correction being considered.

Many models have been proposed to describe the process by which fake news spreads over OSNs. They can be classified into models based on the user's own characteristics [4, 5] and models based on the environment surrounding the user [6–9]. However, since these models deal with the spread of one type of information, they fail to describe the interaction between fake news and corrections. Therefore, it is necessary to consider the characteristics of fake news and corrections, and develop an information spreading model that considers their interaction.

In this paper, we consider the individual characteristics of fake news and corrections and their interaction, and propose a model in which the two types of information are intertwined and spread on the OSNs. The proposed model is based on an activator-inhibitor model of the reaction-diffusion system. This model shows that even if the network structure itself is not biased at all, the interaction between fake news and the corrections may result in the formation of OSN clusters that believe in fake news. This result can give an explanation of the mechanism whereby the corrections amplify the influence of fake news.

The rest of this paper is organized as follows. In Sect. 2, we describe the reaction-diffusion equation for describing the nature and interaction of fake news and the corrections, and in particular, the activator-inhibitor model. In Sect. 3, we give an activator-inhibitor model on OSNs and show how this model can

handle the interpretations of both fake news and corrections. Section 4 performs a numerical experiment on the activator-inhibitor model assuming a homogeneous network model that starts without any cluster structure. The results show that users who are strongly affected by fake news form many clusters in OSNs by the interaction of fake news and corrections even if the OSN structure is not biased. Finally, we will summarize our finding in Sect. 5.

2 Reaction-Diffusion System

The reaction-diffusion systems are mathematical models that describe phenomena in which multiple objects are distributed in a certain space and diffuse in the space while interacting with each other. The reaction-diffusion equations can describe such phenomena. For example, consider two types of objects in a two-dimensional Euclidean space represented by the coordinates (x, y) . Let the density functions representing the spatial distribution of the two types of objects at time t be $a(x, y; t)$ and $h(x, y; t)$, respectively. At this time, the reaction-diffusion equations are given as follows:

$$\begin{aligned}\frac{\partial a(x, y; t)}{\partial t} &= f(a, h; t) + \kappa_a \Delta a(x, y; t), \\ \frac{\partial h(x, y; t)}{\partial t} &= g(a, h; t) + \kappa_h \Delta h(x, y; t),\end{aligned}\tag{1}$$

where, $f(a, h; t)$ and $g(a, h; t)$ in the first terms on the right hand sides are called reaction terms. Since the reaction terms are functions of $a(x, y; t)$ and $h(x, y; t)$, they describe the effect of the interaction between the two types of objects. In particular, because we consider local interactions that are locally determined at each point (x, y) , the reaction terms are determined by the information at the point

$$\begin{aligned}f(a, h; t) &= f(a(x, y), h(x, y); t), \\ g(a, h; t) &= g(a(x, y), h(x, y); t).\end{aligned}\tag{2}$$

The second terms on the right hand sides are called the diffusion terms, they describe how the two types of objects diffuse spatially. Δ denotes the Laplacian

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2},$$

and κ_a and κ_h are diffusion coefficients that describe the intensity of the diffusion of each object. The larger the diffusion coefficient, the faster is the diffusion.

The reaction-diffusion system is widely applied not only in the field of chemistry but also in other fields such as biology, ecology, and physics. The activator-inhibitor model, which is a type of reaction-diffusion system, has been applied also in network engineering [11, 12]. These application examples utilize the Turing pattern that is demonstrated by the activator-inhibitor model. The Turing

pattern is a kind of spatial pattern that is spontaneously generated from the system and is known to explain the generation mechanism of the geometric patterns that can appear on the surfaces of animal bodies.

In the activator-inhibitor model used in [11, 12], the reaction terms of the reaction-diffusion Eq. (1) are determined as follows:

$$\begin{aligned} f(a, h; t) &= \frac{c a(x, y; t)}{h(x, y; t)} - \mu_a a(x, y; t) + \rho_a, \\ g(a, h; t) &= c a(x, y; t) - \mu_h h(x, y; t) + \rho_h, \end{aligned} \quad (3)$$

where c , μ_a , μ_h , ρ_a , and ρ_h are positive constants. In addition, as a more simple and natural way of giving the reaction terms, we can use the following approach:

$$\begin{aligned} f(a, h; t) &= c_a a(x, y; t) - c_h h(x, y; t) \\ &\quad - \mu_a a(x, y; t) + \rho_a, \\ g(a, h; t) &= c a(x, y; t) - \mu_h h(x, y; t) + \rho_h, \end{aligned} \quad (4)$$

where c_a and c_h are positive constants.

In the activator-inhibitor model, $a(x, y; t)$ is called the activator and $h(x, y; t)$ is called the inhibitor. The intent of this naming can be understood by looking at the structure of the reaction term in (3) or (4). The meaning of the reaction terms can be explained qualitatively as follows. The presence of the activator acts to promote the increase of both the activator and the inhibitor at that location, and the presence of the inhibitor acts to suppress the increase of the activator at that location. Figure 1 illustrates this relationship. The second terms on the right hand sides of (3) and (4) represent the reaction of self-decay, and the third terms on the right hand sides of (3) and (4) represent the rate of natural generation.

Next, with regard to the diffusion terms of the reaction-diffusion equations (1), the activator-inhibitor model requires the following relationship

$$\kappa_h > \kappa_a, \quad (5)$$

meaning that the inhibitor spreads faster and more widely than the activator.

In the activator-inhibitor model, when the reaction term and the diffusion term work appropriately together, the following situations occur.

- The activator present at a position proliferates both the activator and the inhibitor at the same position.
- At that time, the inhibitor quickly diffuses around the position and suppresses the increase of the activator in the surroundings.
- When the activator is suppressed in the surroundings, not only the activator but also the growth of the inhibitor is suppressed.
- When the inhibitor decreases, both the activator and the inhibitor proliferate by the action of the activator, and the generated inhibitor diffuses to its surroundings more quickly than the activator.

By repeating such a process, the spatial distributions of the activator and the inhibitor become uneven, and appear as geometric patterns in the space. This is called the Turing pattern. Figure 2 shows an example of the evaluation of the behavior of the activator and the inhibitor by the activator-inhibitor model in one-dimensional space [13]. We can recognize that a periodic pattern appears in the spatial distributions of both the activator and the inhibitor with the passage of time.

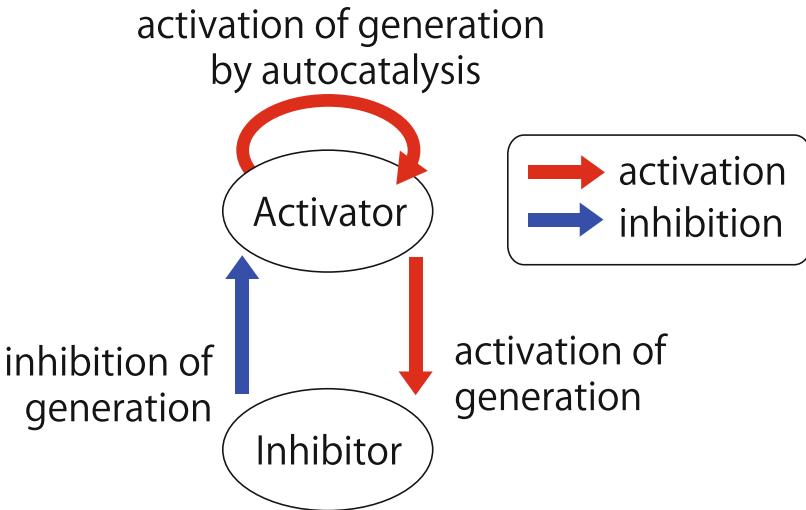


Fig. 1. Basic mechanism of the activator-inhibitor system

3 Activator-Inhibitor Model for Fake News and Corrections

3.1 Basic Concept

Based on the case of hoarding toilet paper shown in Sect. 1, we propose a mathematical model to understand the mechanism by which the spread of both fake news and the corrections results in toilet paper hoarding. In preparation, let us consider the basic properties of fake news and the corrections and their interaction. We assume the following properties as a starting point for the model.

- In general, fake news deals with topics that are of interest to users and therefore tend to spread faster over OSNs than regular news [10].
- The spread of the corrections fuels users' anxiety that toilet paper may be bought up by many people contrary to the original purpose of the corrections. For this reason, the corrections not only convey their original intent, but also attracts attention to the fake news.

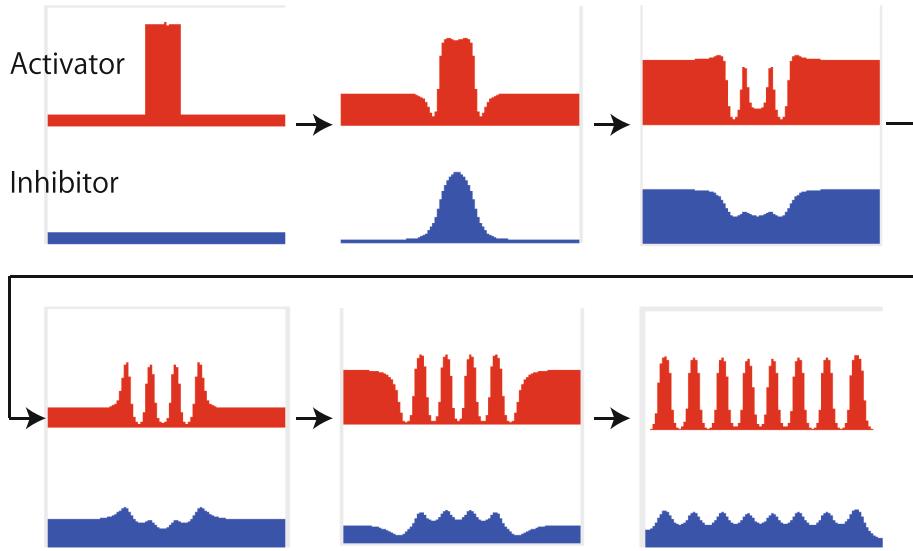


Fig. 2. Turing patterns generated by the activator-inhibitor system

- The spread of fake news makes the content of fake news known, and it becomes difficult to spread the corrections that deny it.

Based on the above properties, the following correspondence can be assumed in the activator-inhibitor model.

- activator = corrections
- inhibitor = fake news

Figure 3 shows the structure of this relationship in contrast to Fig. 1.

3.2 Activator-Inhibitor Model on Networks

In this section, we propose an activator-inhibitor model that describes the interaction between fake news and corrections in OSNs, based on the concept shown in Sect. 3.1.

Consider undirected graph $G(V, E)$ with n nodes representing OSN structure, where $V = \{1, 2, \dots, n\}$ is the set of nodes and E is the set of undirected links. In terms of correspondence with OSN, node $i \in V$ represents a user, and undirected link $(i, j) \in E$ represents the relationship between user i and user j .

Originally, the strength of relationships between users in OSNs is asymmetric, so it is natural to model the structure of OSNs with a directed graph. However, in this paper, we would like to discuss the phenomenon that information dispersal becomes biased even though the network structure itself is not biased at all. For this reason, we dare to introduce a highly symmetric network model represented by an undirected graph. As a side note, it is easy to extend the network model

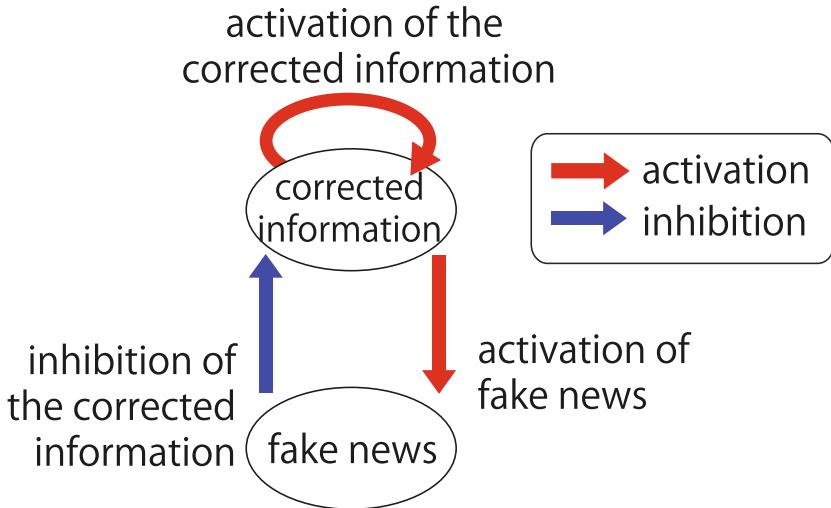


Fig. 3. Interaction between fake news and their corrections

to directed graphs and to discuss information spreading on them, as there are no technical problems in doing so.

We assume that each link in $G(V, E)$ has a link weight that represents the strength of the relationship, and let the weight of undirected link $(i, j) \in E$ between node i and node j be $w_{ij} > 0$. For convenience, the weight of the node pair $(i, j) \notin E$ is assumed to $w_{ij} = 0$.

The Laplacian matrix \mathbf{L} representing the structure of $G(V, E)$ is defined as

$$\mathbf{L} := \mathbf{D} - \mathbf{A}, \quad (6)$$

where \mathbf{A} is the adjacency matrix defined as

$$\mathbf{A} = [w_{ij}]_{1 \leq i, j \leq n},$$

and \mathbf{D} is the weighted degree matrix defined as

$$\mathbf{D} = \text{diag}(d_1, d_2, \dots, d_n).$$

Here, d_i is the weighted nodal degree of node i defined as

$$d_i = \sum_{j=1}^n w_{ij}.$$

The Laplacian matrix \mathbf{L} can be used to describe the diffusion phenomenon on the network. Let $x_i(t)$ (≥ 0) be the state quantity of node i at time t , and the state vector $\mathbf{x}(t)$ be defined as

$$\mathbf{x}(t) := {}^t(x_1(t), x_2(t), \dots, x_n(t)).$$

The diffusion equation on the network can, by using the Laplacian matrix, be expressed as

$$\frac{d}{dt} \mathbf{x}(t) := -\mathbf{L} \mathbf{x}(t).$$

When considering the activator-inhibitor model on the network, the diffusion term appearing in the activator-inhibitor model can be expressed using the Laplacian matrix. The activator-inhibitor model on the network is expressed as follows. Let $a_i(t)$ and $h_i(t)$ be the amounts of activator and inhibitor at node i at time t , respectively. Next, we define two types of state vectors,

$$\begin{aligned} \mathbf{a}(t) &:= {}^t(a_1(t), a_2(t), \dots, a_n(t)), \\ \mathbf{h}(t) &:= {}^t(h_1(t), h_2(t), \dots, h_n(t)), \end{aligned} \quad (7)$$

which have the amounts of activators and inhibitors for all nodes as elements. Then, the equations that describe the activator-inhibitor model on the network can be written as

$$\begin{aligned} \frac{\partial}{\partial t} \mathbf{a}(t) &:= \mathbf{F}(\mathbf{a}(t), \mathbf{h}(t)) - \kappa_a \mathbf{L} \mathbf{a}(t), \\ \frac{\partial}{\partial t} \mathbf{h}(t) &:= \mathbf{G}(\mathbf{a}(t), \mathbf{h}(t)) - \kappa_h \mathbf{L} \mathbf{h}(t). \end{aligned} \quad (8)$$

Here, $\mathbf{F}(\mathbf{a}(t), \mathbf{h}(t))$ and $\mathbf{G}(\mathbf{a}(t), \mathbf{h}(t))$ are functions that express the reaction terms, and $-\kappa_a \mathbf{L} \mathbf{a}(t)$ and $-\kappa_h \mathbf{L} \mathbf{h}(t)$ are the diffusion terms. By following (4), the functions representing the reaction terms are introduced as

$$\begin{aligned} \mathbf{F}(\mathbf{a}(t), \mathbf{h}(t)) &:= \mathbf{C}_a \mathbf{a}(t) - \mathbf{C}_h \mathbf{h}(t) \\ &\quad - \mathbf{M}_a \mathbf{a}(t) + \mathbf{R}_a, \\ \mathbf{G}(\mathbf{a}(t), \mathbf{h}(t)) &:= \mathbf{C}_h \mathbf{a}(t) - \mathbf{M}_h \mathbf{h}(t) + \mathbf{R}_h. \end{aligned} \quad (9)$$

Here, \mathbf{C}_a , \mathbf{C}_h , \mathbf{M}_a , \mathbf{M}_h , \mathbf{R}_a , and \mathbf{R}_h are all diagonal matrices defined as

$$\begin{aligned} \mathbf{C}_a &:= c_a \mathbf{I} = \text{diag}(c_a, \dots, c_a), \\ \mathbf{C}_h &:= c_h \mathbf{I} = \text{diag}(c_h, \dots, c_h), \\ \mathbf{M}_a &:= \mu_a \mathbf{I} = \text{diag}(\mu_a, \dots, \mu_a), \\ \mathbf{M}_h &:= \mu_h \mathbf{I} = \text{diag}(\mu_h, \dots, \mu_h), \\ \mathbf{R}_a &:= \rho_a \mathbf{I} = \text{diag}(\rho_a, \dots, \rho_a), \\ \mathbf{R}_h &:= \rho_h \mathbf{I} = \text{diag}(\rho_h, \dots, \rho_h), \end{aligned}$$

where \mathbf{I} is an $n \times n$ unit matrix. The fact that these matrices are diagonal matrices reflects the assumption that the interaction between the activator and the inhibitor is a local interaction at each node.

In the above model, for the sake of simplicity and higher symmetry, all the above diagonal matrices are set to be proportional to the identity matrix. This corresponds to all nodes having the same parameter value. More generally, we can use a diagonal matrix where each node has a different parameter value.

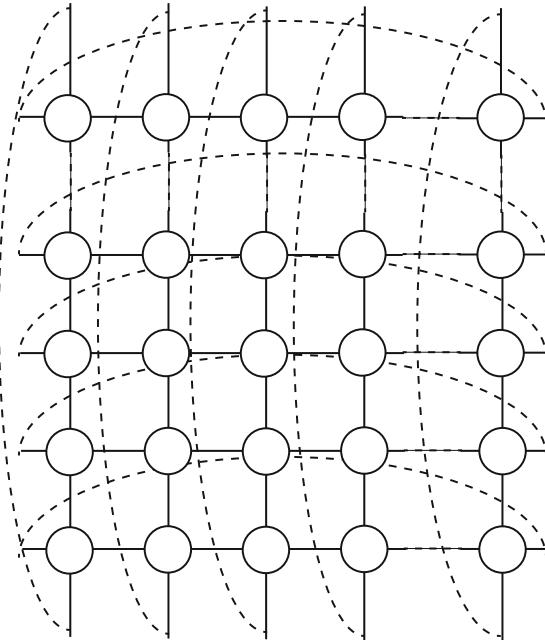


Fig. 4. 100×100 2-dimensional lattice network model with torus boundary

Table 1. Parameter setting for the activator-inhibitor model

c_a	c_h	μ_a	ρ_a	κ_a	c	μ_h	ρ_h	κ_h
0.08	0.08	0.03	0.2	0.02	0.11	0.06	0.15	0.5

4 Experimental Evaluation

This section performs numerical experiments of the activator-inhibitor model that describes the interaction between fake news and the corrections on a homogeneous network with higher symmetry that does not have a cluster structure. The purpose of the experimental evaluations is to show that even if there are no clusters in the network structure itself, clusters of users that are strongly influenced by fake news might be formed on OSNs. In the experiment, (4) was adopted as the reaction term of the activator-inhibitor model. The experiments were conducted by using the Reaction-Diffusion System Simulator [14] published on the website of Professor Shigeru Kondo's Laboratory, Osaka University.

The experimental environment is described as follows. We use a two-dimensional lattice network model consisting of 10,000 nodes in a 100×100 configuration, with a periodic boundary condition that yields a torus topology (Fig. 4). All links are weighted at 1. Other parameters are summarized in Table 1. In addition, the maximum possible values of $a_i(t)$ and $h_i(t)$ are set to 0.2 and 0.5, respectively.

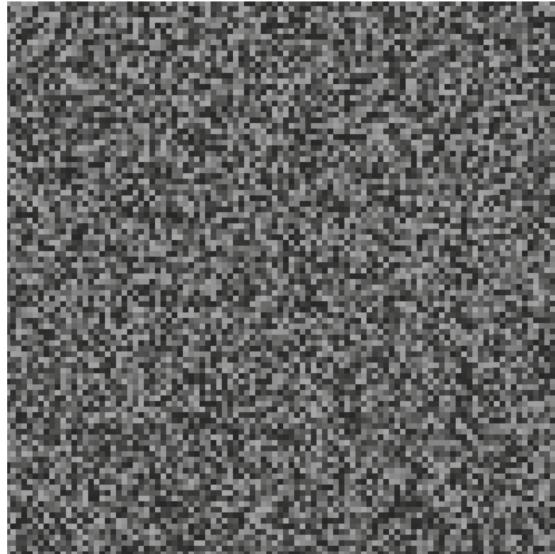


Fig. 5. The initial condition of the correction information (activator)

Figures 5, 6 show the initial condition and the evaluation result, respectively. The initial values of activator $a_i(0)$ and inhibitor $h_i(0)$ are randomly given for each node i as the initial state at $t = 0$. Figure 5 shows the initial distribution of the activator for each node placed the 100×100 node configuration. Figure 6 shows the distribution of activators after sufficient time has passed for each node in the 100×100 node configuration. The dark part indicates where the activator value is large, and the light part indicates where the activator value is small. The difference in activator value directly corresponds to the difference in inhibitor value. That is, where the value of the activator is large, the value of the inhibitor is also large. Conversely, where activator value is small, the inhibitor value is also small. From the correspondence with fake news, we find that fake news is also activated in areas where the corrections are actively discussed, while fake news is not attracting attention in places where the corrections do not receive attention. The important point of this experimental result is that the corrections do not uniformly reduce the influence of fake news, but causes differentiation yielding parts where fake news is strongly noticed and parts where it is not. Since the network model used in the experiment is a network model with very high symmetry (a homogeneous undirected graph with lattice topology) and periodic boundary conditions, the network structure has not location-specific properties. Therefore, the polarization that appears in the experimental result is the heterogeneity created by the interaction of fake news and the corrections. In this way, the activator-inhibitor model can qualitatively explain the phenomenon in which the corrections cause a situation in which fake news receives more attention from users.

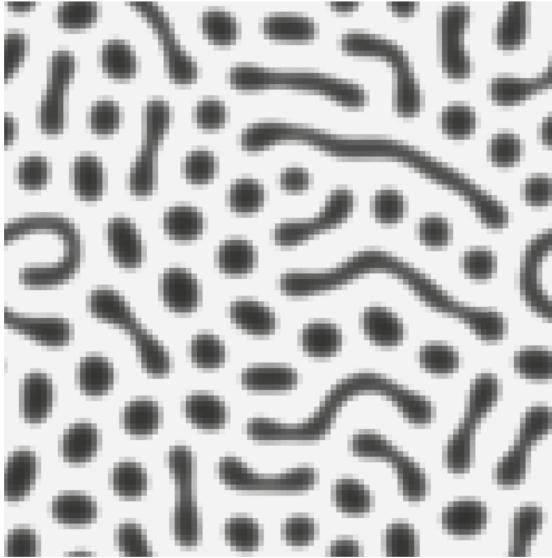


Fig. 6. Turing patterns generated by the interaction between fake news and their corrections

5 Conclusion

In this paper, we proposed a model that explains the occurrence of the phenomenon in which the information issued to correct fake news on OSNs does not always yield the expected effect, and may in fact, worsens the influence of fake news. By using the activator-inhibitor model, which is a kind of reaction-diffusion system, and making the inhibitor correspond to fake news and the activator correspond to the corrections, we were able to well replicate the characteristics and interactions of both. Numerical experiments showed that even if there is no bias in the network structure itself, the interaction between fake news and the corrections may trigger the generation of user clusters that are strongly influenced by fake news. This result provides essential insights into the development of strategies that can well counter fake news. For example, if the corrections spread faster over OSNs than fake news, the inequality (5) does not hold. Therefore user clusters as the Turing patterns will be disappeared. Thus, making the corrections spread faster is one of the hopeful countermeasures for fake news. In the future, based on the proposed activator-inhibitor model, we will examine effective and strategic methods for transmitting the corrections as a countermeasure against fake news.

Acknowledgement. The authors would like to thank Mr. Satoshi Furutani at NTT Labs. for useful discussions. This research was supported by Grant-in-Aid for Scientific Research (B) No. 19H04096 (2019–2021), No. 20H04179 (2020–2022), and No. 21H03432 (2021–2023), and Grant-in-Aid for Challenging Research (Exploratory) No. 21K19775 (2021–2023) from the Japan Society for the Promotion of Science (JSPS), and TMU local 5G research support.

References

1. Guess, A., Nyhan, B., Reifler, J.: Selective exposure to misinformation: evidence from the consumption of fake news during the 2016 U.S. presidential campaign. European Research Council (2018)
2. Toriumi, F.: Study Group on Platform Service. Ministry of Internal Affairs and Communications, Japan, April 2021. https://www.soumu.go.jp/main_content/000745175.pdf
3. Nikkei Business Online. <https://business.nikkei.com/atcl/gen/19/00002/022801115/>
4. Brody, D.C., Meier, D.M.: How to model fake news. [arXiv:1809.00964](https://arxiv.org/abs/1809.00964) (2018)
5. Causal understanding of fake news dissemination on social media. [arXiv:2010.10580](https://arxiv.org/abs/2010.10580) (2020)
6. Törnberg, P.: Echo chambers and viral misinformation: modeling fake news as complex contagion. PLoS One **13**, 1–21 (2018)
7. Peng, H., Nematzadeh, A., Romero, D.M., Ferrara, E.: Network modularity controls the spread of information diffusion. Phys. Rev. E **102**, 052316 (2020)
8. Lotito, Q.F., Zanella, D., Casari, P.: Realistic aspects of simulation models for fake news epidemics over social networks. Future Internet **13**(3), 76 (2021)
9. Hébert-Dufresne, L., Scarpino, S.V., Young, J.-G.: Macroscopic patterns of interacting contagions are indistinguishable from social reinforcement. Nat. Phys. **16**, 426–431 (2020)
10. Vosoughi, S., Roy, D., Aral, S.: The spread of true and false news online. Science **359**(6380), 1146 (2018)
11. Neglia, G., Reina, G.: Evaluating activator-inhibitor mechanisms for sensors coordination. In: Proceedings of International Conference on Bio-Inspired Models of Network, Information and Computing Systems (BIONETICS 2007), pp. 129–133 (2007)
12. Wakamiya, N., Hyodo, K., Murata, M.: Reaction-diffusion based topology self-organization for periodic data gathering in wireless sensor networks. In: Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems, pp. 351–360 (2008)
13. Aida, M.: Distributed Control and Hierarchical Structure in Information Networks. Information Network Science Series, vol. 3, Corona Publishing Co. Ltd. (2015). (in Japanese)
14. Reaction-Diffusion System Simulator. Kondo Laboratory, Osaka University. www.fbs.osaka-u.ac.jp/labs/skondo/simulators/rd_new.html



Love and Hate During Political Campaigns in Social Networks

Juan Carlos Losada¹, José Manuel Robles², Rosa María Benito¹,
and Rafael Caballero^{2(✉)}

¹ Complex System Group, Universidad Politécnica de Madrid, Madrid, Spain
{juancarlos.losada,rosamaria.benito}@upm.es

² Universidad Complutense de Madrid, Madrid, Spain
jmrobles@ccee.ucm.es, rafacr@ucm.es

Abstract. Social networks have become central for public debate. However, this debate occurs, in many cases, in a polarized and socially fragmented way. The importance of a specific type of polarization marked by the expression of emotions (affective polarization) has recently been pointed out. This work seeks to analyze affective polarization through the study of opinions shared by users of Twitter during competitive events such as a political election (United States electoral campaign, 2016). To operationalize the affective polarization we propose, first, to consider the relationship between the user opinions about each contender. This approach allows us to describe what is the opinion of the supporters of one contender about the rest of them. Secondly, we combine sentiment analysis techniques, new diagrams describing graphically the tension between the positive and negative opinions, and numerical measurements obtained employing techniques borrowed from physics such as the gravity centers to identify and measure affective dispositions of users participating in this discussion. In summary, a way of measuring an emergent and central type of polarization in competitive contexts is proposed here.

Keywords: Social networks · Sentiment analysis · Love-hate diagram

1 Introduction

Initially proposed by Barnes [1] in the field social science, the concept of *social network* acquired a strong impulse with the raise of Internet, and in particular with the creation of social network web sites. According to Ellison *et al.* [2], a social network web site (Twitter in our case) is a web-based service that allow individuals to publish information under a public or semi-public profile, manage the list of users with whom they are connected, and check information shared by other users.

Social networks become particularly active during competitive events, where several contenders struggle to achieve some goal. Examples of such events include political processes [3–5], sport championships [6], or even song contests [7].

In these events, the users post their messages, often expressing their support or disagreement about each candidate. These opinions contain valuable information that can contribute to a better understanding of the process. In order to analyse the large amount of user opinions, techniques such as *sentiment analysis* [8] are employed. These techniques generate values representing the opinion of a post about a particular subject, in our case about each candidate. In our paper we assume that each post can be labeled as -1 (negative, which we call sometimes ‘hate’), 0 (neutral) or $+1$ (positive also called in this paper ‘love’), obtaining a different label for each candidate. Next, the message polarities from the same user are averaged to obtain the user opinion about each contender.

This process yields information about the support of each contender in the social network. However, this says little about the *polarization* of the debate, which constitutes an important question in the academic literature [9]. For this reason, this work proposes a methodological instrument devoted to analyze the level of polarization in these events.

In this paper we focus on the case of events with two contenders, in order to adequately describe the methodology, but it can be easily generalized to processes with more than two participants. The next section shows the diagrams and measurements that we propose to study this process. Then, in Sect. 3 we apply these tools to the particular case of the USA 2016 elections in Twitter. The paper ends showing some conclusions and future work.

2 Detecting Love and Hate

We are interested in inferring the opinion that Twitter users have towards two candidates in an electoral process, which we will call candidates C and T due to similarity to the case study that will be presented later.

At the top of the Fig. 1 is shown the process of assigning sentiment values to users. We assign a value X_i to the user u_i , (where X can be T or C) given by the average of the opinion values assigned to X in each message of u_i . The values $X_i = -1$ indicating totally negative sentiment or hate, $X_i = +1$ totally positive or love, and $X_i = 0$ a neutral sentiment for the candidate X.

We call U^X the set of users with sentiment for candidate X. In general, there may be users who never post about a candidate and also messages where the users post about both. This implies that, in general, the sets U^C and U^T have a different number of elements. We denote by $U^{CT} = U^C \cap U^T$ the set of users with sentiment for both candidates. These users can be represented in a two-dimensional space, as shown in the bottom of Fig. 1.

For each set of user, U^X , we calculate the probability distribution $p(x)$, considering the probability of finding a user with a given sentiment x .

$$p(x) = \frac{\#X_i = x}{\#X_i} \quad (1)$$

From this distribution of sentiments we obtain the polarization indices described in the Ref. [10].

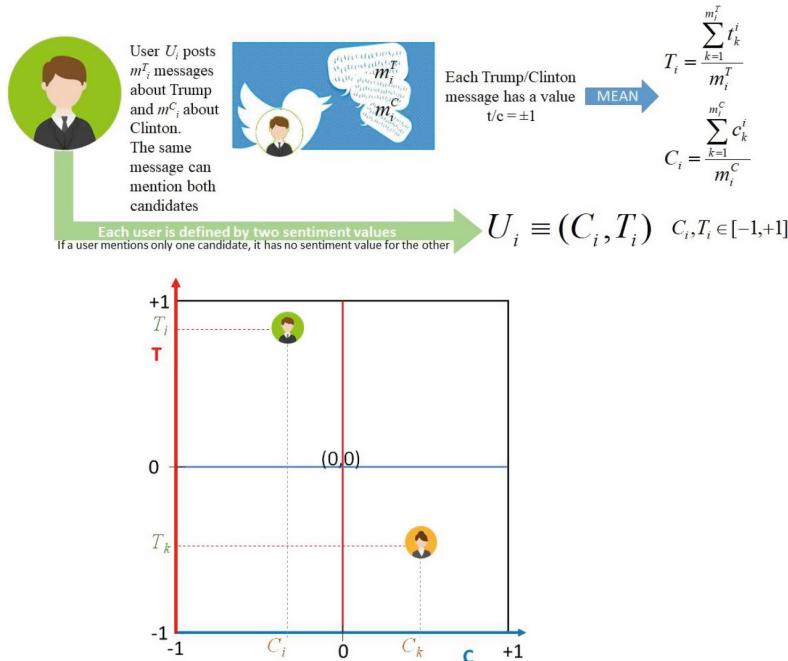


Fig. 1. Top: Construction scheme of user sentiment about C and T candidates. Bottom: Mapping of user sentiment values about both candidates in the (C, T) space.

The proportion of users in each interval of sentiments, $x < 0$ and $x > 0$, is given respectively by:

$$\begin{aligned} A^- &= \int_{-1}^0 p(x) dx \\ A^+ &= \int_0^1 p(x) dx. \end{aligned} \quad (2)$$

The difference between these indices

$$\Delta A = A^+ - A^- \quad (3)$$

gives information on which sentiment is predominant about candidate X.

The average value of the sentiments of love (gc^+) or hate (gc^-) gives information on how extreme these sentiments are. These values can be obtained by calculating the centers of gravity of the function $p(x)$ in each interval:

$$\begin{aligned} gc^- &= \frac{\int_{-1}^0 p(x)x dx}{A^-} \\ gc^+ &= \frac{\int_0^1 p(x)x dx}{A^+}. \end{aligned} \quad (4)$$

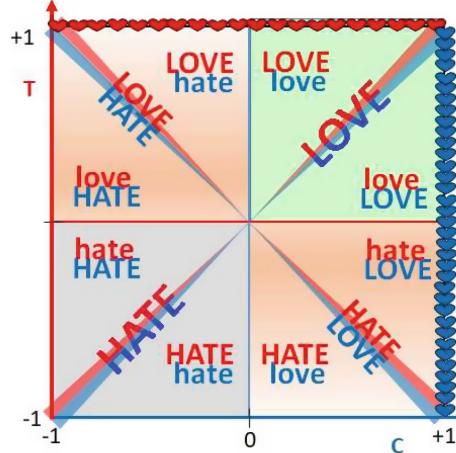


Fig. 2. Structure of a love-hate diagram for two candidates C and T. The axes that represent the maximum love for each candidate, is represented by blue hearts ($C = +1$) and by red hearts ($T = +1$).

A mean value close to ± 1 implies extreme positive/negative sentiments, while if the mean value is close to 0, the sentiments are more neutral.

From these indices, we can define a global polarization index using the following expression:

$$\mu = (1 - |\Delta A|)d, \quad (5)$$

where

$$d = \frac{|gc^+ - gc^-|}{2} \quad (6)$$

it is half the distance between the centers of gravity. In this way μ is a magnitude between +1 and 0, where +1 implies extreme polarization and 0 minimum polarization.

In the case of users who have sentiment towards the two candidates ($u_i \in U^{CT}$), we can also analyze the distribution of sentiments in two-dimensional space (C, T), obtaining what we have called the love-hate diagram, whose scheme is shown in the Fig. 2. Each point in this diagram can be colored based on the number of users with that pair of sentiments. The users located on the line $X = +1$ are those who love candidate X, marked in blue in Fig. 2 for the case $X = C$ and in red for $X = T$.

In the quadrant $C > 0$ and $T > 0$ users have feelings of love towards both candidates, while in the quadrant with $C < 0$ and $T < 0$ users have feelings of hatred towards both of them. On the line with slope +1, there are users who love or hate both candidates in the same way. In polarized situations it is expected that these two quadrants are practically empty, since users do not usually have the same type of feeling towards the two options.

In the quadrant with $C > 0$ and $T < 0$ are the users who love C and hate T. On the line with slope -1 are the users who love C the same as they hate T. By below this line users hate T more than they love C and above are users who love C more than they hate T. Analogously in the quadrant with $C < 0$ and $T > 0$ there are users who love T and hate C. On the line with slope -1 are those users who love T the same as they hate C. Below this line, users hate C more than they love T and above, they love T more than they hate C.

3 A Case Study: The 2016 USA Elections

3.1 The Dataset

As usual in social network research, our study adopts an observational (also known as correlational or non-experimental) rather than an experimental approach point of view. That is, we just collect and analyse data that already exists. We consider two criteria to characterize the subset of the data from the social network that can be relevant for our experiment:

Temporal Criterion. The fact that these events usually have a deadline, establishes an upper temporal bound for the period of time we wish to consider. On the contrary, there is no rule of thumb about the lower temporal bound, that must be determined arbitrarily. In practice, filtering by this criterion is often allowed by the social media application program interfaces. In our case we downloaded tweets in starting at 0 h (UTC) of 2016-11-02 and ending at 9 am (UTC) of 2016-11-13.

Topic Criterion. We aim at collection messages related to our event, but the particular way of filtering the messages will depend on the particular network. In our case we used as topic criterion the Twitter names of both contenders, *@HillaryClinton* and *@realDonaldTrump*, represented by the letters C and T in the rest of the paper.

The initial dataset consisted of 13,358,353 messages or *tweets* following the argot in Twitter, corresponding to 2,967,701 users.

As usual in this kind of research, the raw data needs to be cleansed [11]. In our case, an important phase in our cleaning process involved removing all the users whose language was not English and their tweets, since we chose a training set of English tweets for our polarity classifier. For the sake of brevity we do not discuss here the details, but all the work flow scripts and a documents detailing each step, together with the initial dataset, can be found at <https://rafaelcaballero.github.io/projects/electionsUSA16/>. After this phase we obtained a working dataset of 10,638,997 tweets and 1,937,854 users.

Of these users, 906,422 posted messages including only *@realDonaldTrump*, 491,402 only mentioned *@HillaryClinton* and 540,030 mentioned both. Figure 3 depicts this information graphically in the form of a Venn diagram.

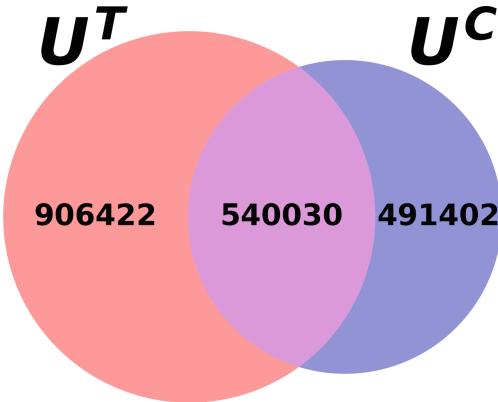


Fig. 3. Venn diagram representing the users that posted messages about each candidate, with U^T representing the set of users mentioning D. Trump, and U^C the set of users mentioning H. Clinton.

3.2 Sentiment Analysis of Tweets

Once we have collected the data, the next step is to classify the messages, according to the opinion they contain toward each contender. Since the number of total messages is expected to be high, we need the help of an automatic or semi-automatic technique, such as sentiment analysis [12]. In our use case, we have decided to use a *naïve Bayes* classifier [13] implemented using the Spark ML library [14].

Although we have tried other more complex classifiers, this simple approach has provided the best results. The main reason is that the messages (tweets) considered in our case are highly unstructured, with little inner grammar complexity. Many messages were in fact lists of words (often campaign terms), used to express their feelings about a contender, and naïve Bayes classifiers work very well in these contexts. Naïve Bayes classifiers are supervised probabilistic classifiers, that is, they provide a probability of belonging to each class. In our case, we have used this feature to mark as neutral (label 0) those tweets with probability under a certain threshold (calculated automatically using cross-validation during the parameter tuning phase). Thus, the classifier distinguishes between negative polarity (represented by label -1), neutral polarity (0), or positive polarity (+1) with respect to the contender. As explained above, we in fact need two models, both based on the same classifier but with different training sets, one for each contender.

The model was trained with 3000 tweets manually labeled for two people. The Cohen's kappa yielded a result of 0.91 agreement in the case of candidate T and 0.88 in the case of candidate C. The trained model was tested on 200 additional tweets manually labeled, giving the following results on precision and recall:

	Precision			Recall		
	-1	0	1	-1	0	1
C	0.80	0.86	0.86	0.76	0.88	0.87
T	0.73	0.81	0.82	0.74	0.88	0.72

The overall F-Score for candidate C is 0.84 and for candidate T 0.78. The results are acceptable, both in precision and recall for both candidates.

3.3 User Opinion About Each Candidate Separately

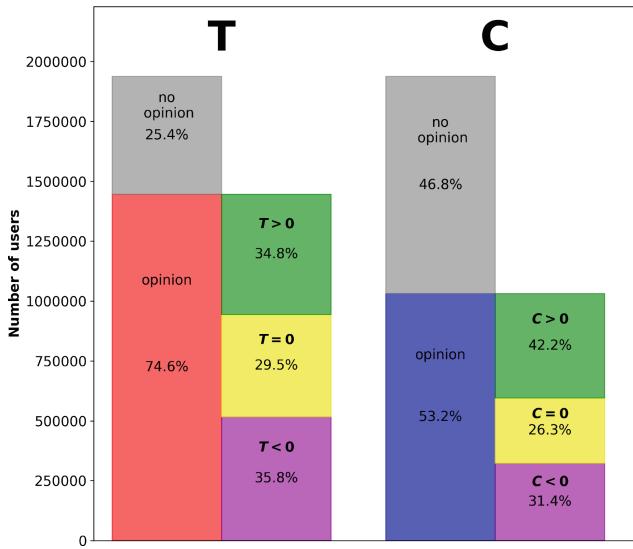


Fig. 4. Distribution of user's opinion for candidate T (left) and C (right). In each case the left-hand side bar represents the proportion of total users with and without opinion about the candidate, while the right-hand side bar divides those with opinion according to their particular sentiment: negative, neutral and positive

From the evaluation of the individual tweets we obtain the average opinion for each user, as explained before. A first glimpse of the results can be observed in Fig. 4. The left hand-side part of the figure shows the distribution of opinions with respect candidate T, while the right-hand side to candidate C. The diagram shows that a 25.4% of the users do not emit any opinion about candidate T, while a 46.8% do not emit any opinion about candidate C.

Among those users with some opinion about C, a majority (42.2%) has a positive opinion, while a 31.4% has a negative opinion and 26.3% are neutral. In the case of candidate T, the proportion of positive opinions is smaller (34.8%),

while the neutral and negative opinion about the candidate is bigger than in the case of candidate C. Thus, we can summarize this information pointing out that:

1. More users emit some opinion about candidate T than about candidate C.
2. In the case of T the neutral and negative opinion represents a bigger proportion than in the case of candidate C.

Thus, if we consider the opinions about each candidate separately we conclude that T obtains more attention than C, but with more negative opinions.

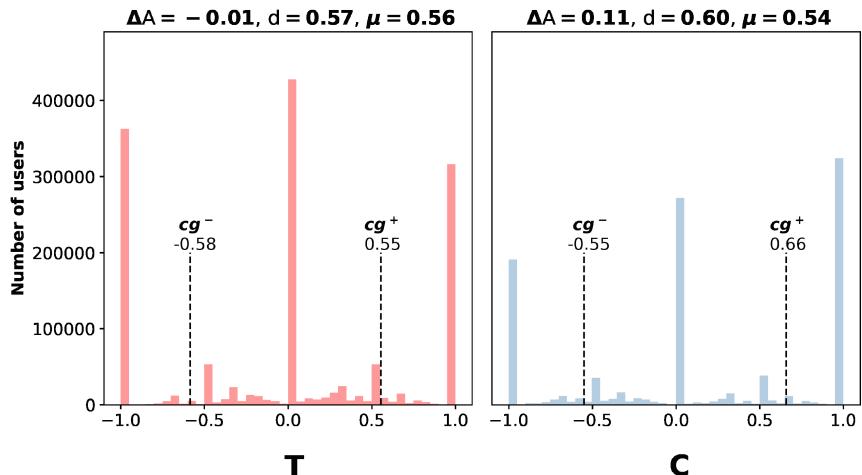


Fig. 5. Distribution of opinions about both candidates, including the gravity centers, the predominant sentiment ΔA , the global polarization index μ and the semi-distance between gravity centers d .

3.4 Polarization

Figure 5 shows the user distributions about each candidate. In the case of candidate T we confirm, as observed above, the negative opinion ($\Delta A < 0$). Moreover, the number of users with strong negative opinion ($T = -1$) is greater than the number of users with extreme positive opinion ($T = +1$). This is confirmed by the gravity center of the negative part which more shifted to the extreme. The overall polarization in the sentiment about candidate T is 0,56.

In the case of candidate C, the positive sentiments are predominant ($\Delta A = +0.11$), and there are more users with extreme positive ($C = +1$) than with extreme negative ($C = -1$) sentiment. In this case we obtain a slightly less polarized context.

However, this is only part of the picture. Indeed, one of the main points of this paper is that this view that considers each candidate separately is not enough, and relating the sentiments about both candidates provides a deeper insight of the situation, as shown in the next sections.

3.5 Relating Opinions About Both Candidates

The Fig. 6 and 7 have two parts:

- The bottom bars contain the same information as Fig. 4, with the difference that in this case the figures represent the proportions of the four parts (negative, neutral, positive and no opinion), instead of separating the proportion of the users with no opinion from the total as in Fig. 4.
- The upper part further shows the opinion about the other candidate assumed that opinion about the first candidate is given in the lower part.

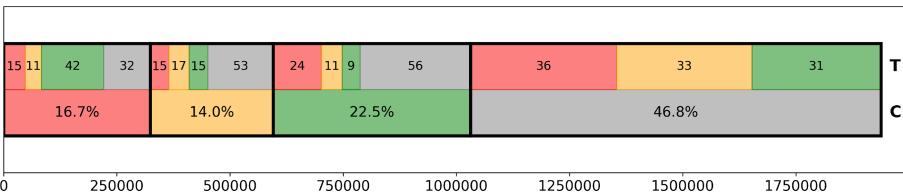


Fig. 6. Distribution of user's opinion for candidate C (below), and each group of opinions in C further divided according to the opinion about T (above). The red color means an overall negative opinion, the yellow color a neutral opinion, the green color a positive opinion and the grey color no opinion about the candidate.

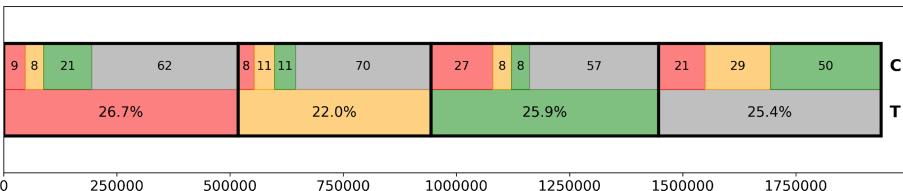


Fig. 7. Distribution of user's opinion for candidate T (below), and each group of opinions in T further divided according to the opinion about C (above). The red color means an overall negative opinion, the yellow color a neutral opinion, the green color a positive opinion and the grey color no opinion about the candidate.

That is, the figures try to show information about questions such as: if we consider the set of supporters of candidate C (green part of the bottom bar in Fig. 6), what opinion do they have about candidate T? Examining the figure we find that:

- As expected, the supporters of each candidate mostly ignore or are against the other candidate, as shown by the sections over the green section of the bottom bars in Figs. 6, 7 and in similar proportion for both candidates.

- In the case of the neutral opinion (yellow bottom bars) both figures show that most of the users do not express an opinion about the other candidate, but this is more accentuate in the case of T, where a 70% of those that present a neutral opinion about T have no opinion about the other candidate.
- Considering the users that have a neutral opinion about one candidate and some opinion about the other (that is, yellow below, not grey above) there are no big differences in the opinion about the second candidate (14% negative, 16% neutral and 15% positive in the case of users with neutral opinion about C, and 8% negative, 10% neutral and 10% positive in the case of those users neutral with respect to T).
- Regarding the users with a negative opinion about one of the candidates, we observe a noticeable difference: of those against T, 62% show no opinion about C, and only a 20% supports the other candidate. However, of those with a negative opinion about C, a 42% support T, the double proportion that in the other case.

Observe that the last point qualifies and even changes the sign of the result of the previous section: while more users express a negative opinion about T, not a great proportion of them consider this disagreement a reason to show support candidate C. On the contrary, those that are against C, show support in a greater proportion to candidate T.

3.6 Love-Hate Diagram

If now we consider the sentiment about each candidate we can depict the love-hate diagram described in Fig. 2, but applied to our particular case. The result can be seen in Fig. 8. Remember that in this diagram support or ‘love’ for T increases to the upper part while support for candidate C increases to the right.

As expected, the quadrants representing users that support both candidates (right upper) contains the smaller number of users, followed very closely by the quadrant associated to those users disagreeing with both candidates (left bottom). Thus, most of the users are in the left upper quadrant (users supporting T and disagreeing with C) and in the right lower quadrant (positive for C, negative for T).

Of these two quadrants clearly the most crowded is the left upper, associated to ‘love’ for T and ‘hate’ for C. Observe how this results presents a new perspective to the result of Sects. 3.3 and 3.4 that examined the opinion about each candidate separately and found that users have a more negative opinion about T than about C. The reason for this difference is that the love-hate diagram considers only users that have opinion about both candidates.

Moreover, we observe that the gravity center of the left upper quadrant is below the diagonal, showing that in the case of users supporting T and disagreeing with C, the ‘hate’ for the opponent is stronger than the ‘love’ for the supported candidate, while in the case of the right lower quadrant the opposite occurs. This confirms the results of Sect. 3.5 in the sense than candidate T seems to obtain more support of those disagreeing with candidate C than the other way round.

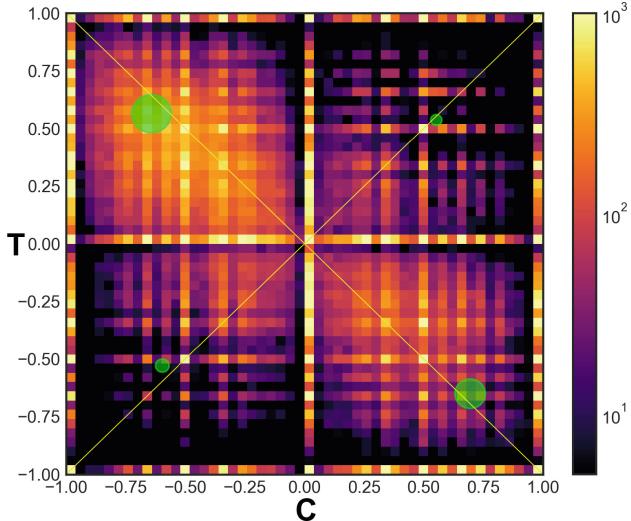


Fig. 8. Love-hate diagram for candidates T and C. The gravity centers of each quadrant are shown as green circles with diameters proportional to the number of users in the quadrant

4 Conclusions

This paper has proposed tools to examine the relationship between the user opinions about two contenders in a competitive context. We think that this is important to get a deeper insight of the process, understanding not only the characteristics of users that support each candidate, but also what is the opinion about the other candidate of these supporters. This allows us to measure the polarization of the event in the social networks, and determine if the disagreement with one candidate implies support to the other candidate.

In particular, we have seen that in the case of the 2016 United States presidential elections, although in a first view users seem to be mostly against candidate Trump, examining the opinions about the two candidates simultaneously we find that this disagreement with Trump does not imply support for Clinton, while this happens to a major extent the other way round. It is worth noticing that our approach can be extended to the case of more than two contenders considering an approach of ‘one against the others’ and comparing the results.

We also think that to have polarization indexes clearly defined can be very valuable when comparing similar events. Thus, an interesting line of future work could be to observe the evolution of the polarization indexes in different contexts:

- During one election, observing if the polarization increases when the election is closer in time.
- Consider different elections and compare the evolution of the polarization indexes.

Acknowledgements. This work has been partially funded by the Spanish research projects PID2019-106254RB-I00 (authors JMR and RC) and PGC2018-093854-B-I00 (authors JCL and RMB) of the Spanish Ministerio de Ciencia Innovación y Universidades of Spain.

References

1. Barnes, J.A.: Class and committees in a Norwegian island parish. *Hum. Relat.* **7**, 39 (1954)
2. Ellison, N.B., et al.: Social network sites: definition, history, and scholarship. *J. Comput.-Mediat. Commun.* **13**, 210 (2007)
3. Bennett, L., Segerberg, A.: The logic of connective action: digital media and the personalization of contentious politics. *Inf. Commun. Soc.* **15**, 739 (2012)
4. Borondo, J., Morales, A.J., Losada, J.C., Benito, R.M.: Characterizing and modeling an electoral campaign in the context of Twitter: 2011 Spanish Presidential election as a case study. *Chaos: An Interdisc. J. Nonlinear Sci.* **22**(2), 023138 (2012)
5. Martín-Gutierrez, S., Losada, J.C., Benito, R.M.: Recurrent patterns of user behavior in different electoral campaigns: a Twitter analysis of the Spanish general elections of 2015 and 2016. *Complexity* **2018**, 2413481 (2018)
6. Kassing, J.W., Sanderson, J.: Fan-athlete interaction and Twitter tweeting through the Giro: a case study. *Int. J. Sport Commun.* **3**, 113 (2010)
7. Highfield, T., Harrington, S., Bruns, A.: Twitter as a technology for audiencing and fandom: the #eurovision phenomenon. *Inf. Commun. Soc.* **16**, 315 (2013)
8. Yadollahi, A., Shahraki, A.G., Zaiane, O.R.: Current state of text sentiment analysis from opinion to emotion mining. *ACM Comput. Surv.* **50**, 1–33 (2017). Article No. 25
9. Fiorina, M.P., Abrams, S.J.: Political polarization in the American public. *Ann. Rev. Polit. Sci.* **11**, 563 (2008)
10. Morales, A.J., Borondo, J., Losada, J.C., Benito, R.M.: Measuring political polarization: Twitter shows the two sides of Venezuela. *Chaos: Interdisc. J. Nonlinear Sci.* **25**, 033114 (2015)
11. Osborne, J.: Best Practices in Data Cleaning: A Complete Guide to Everything You Need to Do Before and After Collecting Your Data. SAGE Publications, Thousand Oaks (2012)
12. Yue, L., Chen, W., Li, X., Zuo, W., Yin, M.: A survey of sentiment analysis in social media. *Knowl. Inf. Syst.* **60**, 617 (2019)
13. Rish, I., et al.: An empirical study of the Naive Bayes classifier. In: IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, vol. 3, pp. 41–46 (2001)
14. Meng, X., et al.: MLlib: machine learning in apache spark. *J. Mach. Learn. Res.* **17**, 1235 (2016)



Homophily - a Driving Factor for Hate Speech on Twitter

Seema Nagar¹(✉), Sameer Gupta², C. S. Bahushruth³, Ferdous Ahmed Barbhuiya¹, and Kuntal Dey¹

¹ Institute of Information Technology, Guwahati, India
seema.nagar@iiitg.ac.in

² National Institute of Technology, Kurukshetra, India
³ Manipal University, Jaipur, India

Abstract. In this paper, we hypothesize that homophily is a powerful driver in the generation of hate speech on Twitter. Prior works have shown the role of homophily in information diffusion, sustenance of online guilds and contagion in product adoption. We observe that familiarity computation techniques used in the literature such as the presence of an edge, and the number of mutual friends between a pair of users, have ample scope of improvement. We address the limitations of existing familiarity computation methods by employing a variational-graph-auto-encoder to capture a user's position in the network and utilizing this representation to compute familiarity. We empirically demonstrate the presence of homophily on a dataset from Twitter. Further, we investigate how homophily varies with different hateful forms, such as hate manifesting in topics of gender, race, colour etc. Our results indicate higher homophily in users associating with topics of racism, sexism and nationalism.

Keywords: Hate speech · Homophily · Social networks

1 Introduction

Hate speech is prevalent on social media platforms. Often, hate speech campaigns on social media platforms have incited real-life violence [12, 15]. Facebook has been blamed for instigating anti-Muslim mob violence in Sri Lanka as well as for playing a leading role in the possible genocide of the Rohingya community in Myanmar by spreading hate speech. Consequently, studying hate speech spread is a fundamental problem with practical implications.

Homophily on social networks was first proposed by [13], using the assortative mixing hypothesis. Homophily is defined as the tendency of like-minded (similar) people to connect/befriend (familiar). A key observation about homophily is that it structures the ego-networks of individuals and impacts their communication behaviour. Therefore, many works have studied the role of homophily in information diffusion and dissemination [3, 6, 9, 17]. Dey et al. [7] detect evidence of homophily in the topic formation and life cycle. Very recently, [19], utilize hashtag based homophily for re-marketing, a practical application with immense effect.

In this paper, we investigate the presence of homophily in hate speech on social media platforms, a new and unexplored facet of hate speech. Familiarity and similarity are the two essential factors in homophily. These two factors are readily available on social media platforms. The feature to become friends or follow other users is leveraged to compute familiarity. While the content produced, profile information or any other meta-data information present is utilized to calculate similarity. The purpose of familiarity computation is to capture familiarity between a pair of users. Existing approaches have used familiarity metrics such edge exists or not, the number of mutual friends or common neighbours, part of the same community/component or not [7].

However, the existing familiarity metrics fail to capture familiarity between a pair of users from the overall position in the social network. Moreover, familiarity metrics is applicable in a situation that has to be established empirically. This is a time consuming, manual, and expensive process. Therefore, we propose using variational graph autoencoder (VGAE) based embeddings of users in the social network to compute familiarity. VGAE based embeddings are shown to be powerful in capturing a vector representation of a user from the position in the graph. The embeddings have proven effective in many social network tasks such as link prediction, node clustering [14], node classification [5], community detection [2], citation recommendation [10].

We use VGAE to encode a user and then use various distance-based metrics such as Cosine, Euclidean, and Manhattan distance to compute familiarity between a pair of users. We use tweets posted by the users to calculate similarity. A user's tweets are concatenated to get a user document. The user document is encoded using an existing text encoder. The text embeddings are then used to compute similarity for a pair of users. We hypothesize that the newly proposed familiarity metrics are effective against the two existing (edge exists or not, mutual friends count). We ask ourselves a question, “*Is homophily exhibited by the users generating hateful content?*”.

Hate speech has multiple forms present, such as hate against race, religion, ethnicity, gender, among others. We believe that the multiple-forms of homophily might exhibit different homophilic behaviour; therefore, we ask ourselves another question, “*Is homophily more prominent for particular forms of hate?*”. We detect various hateful forms present in a corpus using topic modelling. We propose to use latent topic modelling to hateful-forms. We find that particular hateful-forms exhibit stronger homophilic behaviour.

In summary, we make the following contributions:

- We propose a novel way to compute familiarity using graph embedding technique
- We show homophily in hate speech on a dataset from Twitter
- We empirically demonstrate the effectiveness of the newly proposed metrics in establish familiarity against the existing metrics, using homophily as the benchmark of comparison
- We do a deep dive analysis of variations of homophily in different forms of hate

2 Related Work

A large body of works has tried to understand the homophily on social networks. McPherson et al. [13] proposed homophily in social networks first time. This paper

defines homophily exists between two users if they follow each other because of shared common interests. Subsequently, [6] establish that homophily plays a vital role in information diffusion and dissemination on social networks. Their work is built on a critical observation that homophily structures the ego-networks of individuals and impacts their communication behaviour. Another early work, Weng et al. [18] detects the existence of reciprocity that exists on Twitter. They show that users sharing common interests tend to follow each other, therefore, resulting in high reciprocity on Twitter due to homophily. They use similarity and familiarity together, to rank the influential users.

The role of homophily in information diffusion on social networks is further strengthened by [3, 9, 17]. Specifically, [9] investigate homophily in political information diffusion. They proved their hypotheses on two groups, conservative and liberal, and conclude “with homophily, members of the majority group have more network connections and are exposed to more information than the minority group”. Additionally, they also show, “with homophily and a tendency to produce like-minded information, groups are disproportionately exposed to like-minded information, and the information reaches like-minded individuals more quickly than it reaches individuals of opposing ideologies.” Another work by [3] demonstrates homophily in contagion for product adoption on dynamic networks. They empirically establish on the Yahoo IM (instant messenger) system that peer influence is not sufficient to explain contagion in product adoption. They find that homophily is responsible for more than half of the contagion. The role of homophily in sustaining online guilds, especially in the gaming world, is studied by [8]. They find that homophily is an essential indicator of a guild’s sustenance. Thus, homophily has been very well studied in the literature and is very important to explain that many social phenomena are happening in the virtual world.

Many papers have jointly studied familiarity and similarity for modelling solutions on Twitter. Ying et al. [20] show that semantic similarities in trajectories of users on mobile networks are a better indicator for user similarity compare to using the vanilla trajectory similarity. Afrasiabi et al. [1] study communities formed over friendships on the YouTube social network. They observe that communities are formed from similar users on YouTube; however, they do not find large similarity values between friends in YouTube communities. Recently, topical homophily was proposed by [7], where they show the homophily is the driving factor in the emergence of topics and their life cycle. However, the existing literature does not address homophily in hate speech.

3 Central Idea

The proposed approach has three main components:

- Define a novel metric for familiarity computation
- Utilizing the novel metrics in showing homophily in hate speech.
- Detecting hateful forms on social media platforms

In the first component, we use a graph variational auto-encoder to encode a user’s position in the social network. We hypothesize that these graph encodings can capture a user’s position as well as their society from a network’s perspective. The familiarity between a pair of users is computed using these graph encodings. We validate the

newly proposed familiarity metrics by showing homophily in hate speech as part of the second component of the proposed approach. In the third component, we use latent topic modelling to detect multiple hateful forms present in hate speech. We hypothesize that individual hateful forms, differing in nature, might exhibit varied homophilic behaviours.

4 A Novel Familiarity Computation Approach

We aim to capture a user's unified view from a position in their social network. The existing familiarity metrics such as, whether an edge exists or not, mutual friends count, or if users are part of the same community or not, fail to incorporate the unified familiarity for a pair of users. We propose to use a graph encoder, to encode a user's position and then use these encoding vectors to compute familiarity between two users.

4.1 Graph Encoder

The graph encoder is capable of representing a user u' 's latent position in a social network. It takes two inputs, (1) identity matrix and (2) the adjacency matrix, representing the social connectivity. It is trained in an unsupervised manner to learn a vector representation of the input graph. We use a well-established technique called variational graph auto-encoder [11] for our graph encoder. Variational graph auto-encoder (VGAE) constitutes of two parts, (a) the encoder, to encode the graph and (b) the decoder, to decode the graph. The encoder is consist of two layers of graph convolutional layers that encodes a graph into lower dimension latent representation Z . At the same time, the decoder reconstructs the adjacency matrix by optimizing cross-entropy loss between the original and reconstructed adjacency and KL-divergence of the approximate from the true posterior. Figure 1 illustrates the VGAE architecture used.

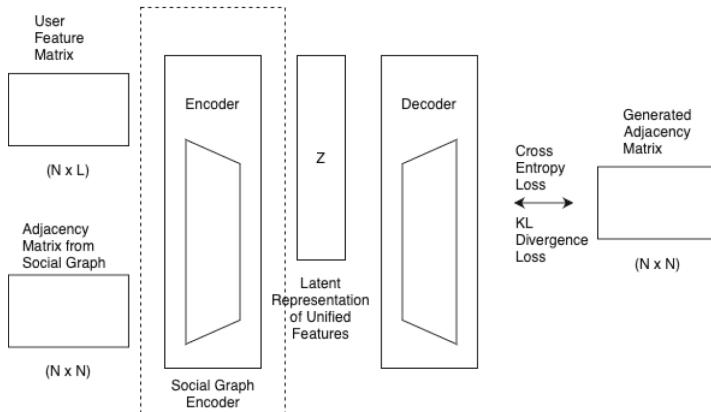


Fig. 1. Variational graph auto encoder

Variational Graph Auto Encoder (VGAE). The encoder takes a data point X and produces a distribution $q\phi(Z|X, A)$. The distribution is usually parameterized as a multivariate Gaussian. Therefore, the encoder predicts the means and standard deviation of the Gaussian distribution. The lower-dimensional embedding Z is sampled from this distribution. The decoder is a variational approximation, $p\theta(A|Z)$, which takes an embedding Z and produces the output \hat{A} .

Encoder or Inferencing Model

The encoder, called graph convolutional network (GCN), consists of two graph convolutional layers. GCN abstracts the social network graph into a latent representation through a convolutional network. The first layer takes adjacency matrix A and the identity matrix as inputs and generates the latent variable Z as output. The first GCN layer generates a lower-dimensional feature matrix \bar{X} , as shown in the formula in Eq. 1 where $\tilde{A} = D^{-1/2}AD^{-1/2}$ is a symmetrically normalized adjacency matrix. The second GCN layer generates μ and $\log \sigma^2$ as shown in Eqs. 2 and 3.

$$\bar{X} = GCN(X, A) = ReLU(\tilde{A}\bar{X}W_0) \quad (1)$$

$$\mu = GCN_\mu(X, A) = \tilde{A}\bar{X}W_1 \quad (2)$$

$$\log\sigma^2 = GCN_\sigma(X, A) = \tilde{A}\bar{X}W_1 \quad (3)$$

The two layers of GCN can be combined as in Eq. 4 and Z can be calculated using Eq. 5, where $\epsilon \in \mathcal{N}(0, 1)$, wherein \mathcal{N} represents a Gaussian (normal) distribution.

$$GCN(X, A) = \tilde{A}.ReLU(\tilde{A}\bar{X}W_0)W_1 \quad (4)$$

$$Z = \mu + \sigma * \epsilon \quad (5)$$

In summary, encoder can be written as in Eq. 6.

$$q(z_i|X, A) = N(z_i|\mu_i, diag(\sigma_i^2)) \quad (6)$$

Decoder

The generative model (decoder) is defined by an inner product between latent variable Z and its transpose Z^T . The output by the decoder (Eq. 8) is reconstructed as adjacency matrix \hat{A} as shown in Eq. 7, where $\sigma()$ is the logistic sigmoid function. In summary, decoder can be represented as in Eq. 8.

$$\hat{A} = \sigma(ZZ^T) \quad (7)$$

$$p(A_{ij} = 1|z_i, z_j) = \sigma(z_i^T z_j) \quad (8)$$

Loss in VGAE

The loss function for variational graph autoencoder has two parts. The first part of the loss function measures how well the network reconstructs the data, and it is called the variational lower bound. We model the loss as binary cross-entropy between the input and the output. The second part of the loss function is KL-divergence between

$q(Z|X, A)$ and $p(Z)$, where $p(Z) = \mathcal{N}(0, 1)$. It measures how closely our $q(Z|X, A)$ matches with $p(Z)$. Formally, the loss can be written as in Eq. 9.

$$L = E_{q(Z|X, A)}[\log p(A|Z)] - KL[q(Z|X, A)||p(Z)] \quad (9)$$

4.2 Computing Familiarity

Familiarity for a pair is computed, using the distance between a pair of users in the graph embedding space. We propose to use existing distance measures to compute familiarity. Distance is inversely proportional to familiarity, lesser the distance, higher the familiarity. Specifically, we experiment with, (a) Cosine distance, (b) Manhattan distance and (c) Euclidean distance to compute familiarity. Formally, the metrics between two users u_1 and u_2 are defined in the Eqs. 10 to 12.

Let \bar{u}_1 and \bar{u}_2 represent graph embedding of the users u_1 and u_2 respectively.

$$\text{CosineSimilarity}(u_1, u_2) = \frac{(\bar{u}_1 \cdot \bar{u}_2)}{\|\bar{u}_1\| \cdot \|\bar{u}_2\|} \quad (10)$$

$$\text{ManhattanDistance}(u_1, u_2) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p} \quad (11)$$

$$\text{EuclideanDistance}(u_1, u_2) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (12)$$

4.3 Hateful Forms Detection Using Topic Modelling

We propose to use topical modelling to detect the hateful forms. We believe the latent topics present are capable of encompassing the hateful forms. Specially, we use LDA [4] to detect the latent topics present in tweets. Due to tweets being short in length and large in number, scaling of LDA to detect topics where every tweet is treated as one document is very challenging. Therefore, we create one document per user by concatenating all his posts, which includes tweets, retweets and quotes. Let a user u_i have made posts P , where $P = p_1, p_2, \dots, p_N$. Then, the document d_i for user u_i is created by concatenating all the tweets in one document. Therefore, we have:

$$d_i = \cup_{(\forall p_j \in P)} p_j \quad (13)$$

Let $D = (\forall i \in 1..n) d_i$ be the corpus of documents. We further investigate D to detect latent topics present in it using LDA based techniques. We explore two variants of sampling for LDA, a) variational Bayes sampling method and b) Gibbs Sampling. Let the set of latent topics is T , where $T = t_1, t_2, \dots, t_n$. The latent topic modelling produces a vector of topic affinity scores v_{d_i} for each document d_i . Let a_i is the affinity scores with respect to topic t_i , then we have topic affinity vector as follows:

$$T_{d_i} = \langle a_1, a_2, a_3, \dots, a_T \rangle \quad (14)$$

5 Experiments

5.1 Experiments Overview

The purpose of the experiments is to investigate the following research questions (RQs) as discussed in Sect. 3:

- RQ1: Is homophily exhibited by the users generating hateful content?
- RQ2: How effective is the newly proposed familiarity metric in comparison to the existing?
- RQ3: Is homophily more prominent for particular forms of hate?

5.2 Experiments Settings

Experimental Dataset. We use hate speech dataset provided by [15]. This dataset contains 200 most recent tweets of 100,386 users, totaling to around $19M$ tweets. It also contains a retweet induced graph of the users. The retweet-induced graph is a directed graph $G = (V, E)$ where each node $u \in V$ represents a user in Twitter, and each edge $(u_1, u_2) \in E$ represents a retweet in the network, where the user u_1 has retweeted user u_2 . Furthermore, every tweet is categorized as an original tweet, retweet, or quote (retweet with a comment). Out of the 100,386 users, labels (hateful or normal) are available for 4,972 users, out of which 544 users are labeled as hateful and the rest as normal. The dataset does not have labels for the tweet content, therefore we manually annotate the tweets as hateful or not. Annotating $19M$ tweets is a very expensive and time-consuming process, hence we annotate a subset of tweets. We manually annotated 30,720 tweets, where the users were picked based on their degree in the retweet network. The filtered retweet network has 7,711,401 edges and 18,642 nodes. The percentage of hateful tweets is 27.5 in the total annotated tweets. We employ pre-processing techniques suitable to tweets. We remove links, convert emoticons to text, and finally remove non ascii characters.

Parameter Setting. We use the source code provided by¹ to implement Variational Graph Auto-Encoder(VGAE). The adjacency matrix is constructed by using the retweet graph as an undirected graph. For training the VGAE, learning rate is set to 0.01, number of epochs are 120, the batch size is the whole graph and Adam optimizer is used. The encoder portion of the trained VGAE is used as graph encoder for further experimentation.

To compute similarity between two users, we use universal sentence encoder (USE)² to encode a user’s tweets. We create a document for a user by concatenating all his tweets and then use USE to encode the document. We use cosine similarity of the documents to get similarity score for a pair of users. Familiarity for a pair is computed using two existing and three newly proposed metrics. The two existing metrics are: edge exists or not and number of mutual friends between a pair of users. The three different novel ways of computing familiarity we implement are cosine similarity, Euclidean and

¹ https://github.com/rusty1s/pytorch_geometric.

² <https://tfhub.dev/google/universal-sentence-encoder/4>.

Manhattan distance on the user vectors obtained from the graph encoder. We use `scipy`³ to compute the similarity and familiarity metrics.

We create a document for each user by combining all the hateful posts (tweets, retweets and quotes). We create a corpus of all the documents of the users. We explored LDA based topic modelling using `MALLET`⁴. We perform grid search where we vary alpha between 0.1 and 0.01 and the number of topics from 6 to 12. The number of iterations for each run is 500. We look at the coherence scores [16] and visualization of topics in terms of overlap using `pyLDAvis`⁵. We find that the best performing topic model, which has both a high coherence score and the least number of overlapping topics, is when $\alpha = 0.1$ and number of topics equal to 8. We observe that $\alpha = 0.01$ gives us a higher coherence score for the same number of topics as compared to $\alpha = 0.1$. From this, we infer that a document in our corpus contains just a few topics.

5.3 Experiments Results

To answer RQ1 and RQ2, we plot similarity against familiarity for the three types of familiarity metrics in Fig. 2. We vary the percentage of hateful tweets for a user and observe the corresponding homophily. We see that as the percentage of hateful tweets increases, similarity also increases. To measure the efficacy of our approach, we compare the Pearson correlation coefficient r for the three approaches. We calculate that the value of r for our approach is 0.6, as compared to 0.5 for edge based familiarity; both indicating a moderate positive relationship. The r value for the mutual friends based familiarity is 0.2, which indicates a weak positive relationship. For the Euclidean and Manhattan distance metrics, as described in Eqs. 12 and 11, the values of r are below 0.1, though greater than 0. Hence, we do not showcase these results and use the Cosine Similarity metric instead, as described in Eq. 10.

Table 1. Top hashtags for the hateful topics

Topic	Hashtags
0	#maga, #trump, #realdonaldtrump, #trumptrain
1	#impeachtrump, #trump, #trumprussia, #jfkfiles
2	#bitch, #metoo, #harvey, #lockherup
3	#gobills, #pelicans, #mlscupplayoffs
4	#london, #fakenews, #cancer, #queen
5	#tormentedkashmir, #kashmirsuffering, #pakistan
6	#brexit, #crime, #terrorism, #illegal
7	#nigga, #bitch, #bitches, #somalia, #nigger

³ <https://www.scipy.org/>.

⁴ <http://mallet.cs.umass.edu/>.

⁵ <https://pypi.org/project/pyLDAvis/>.

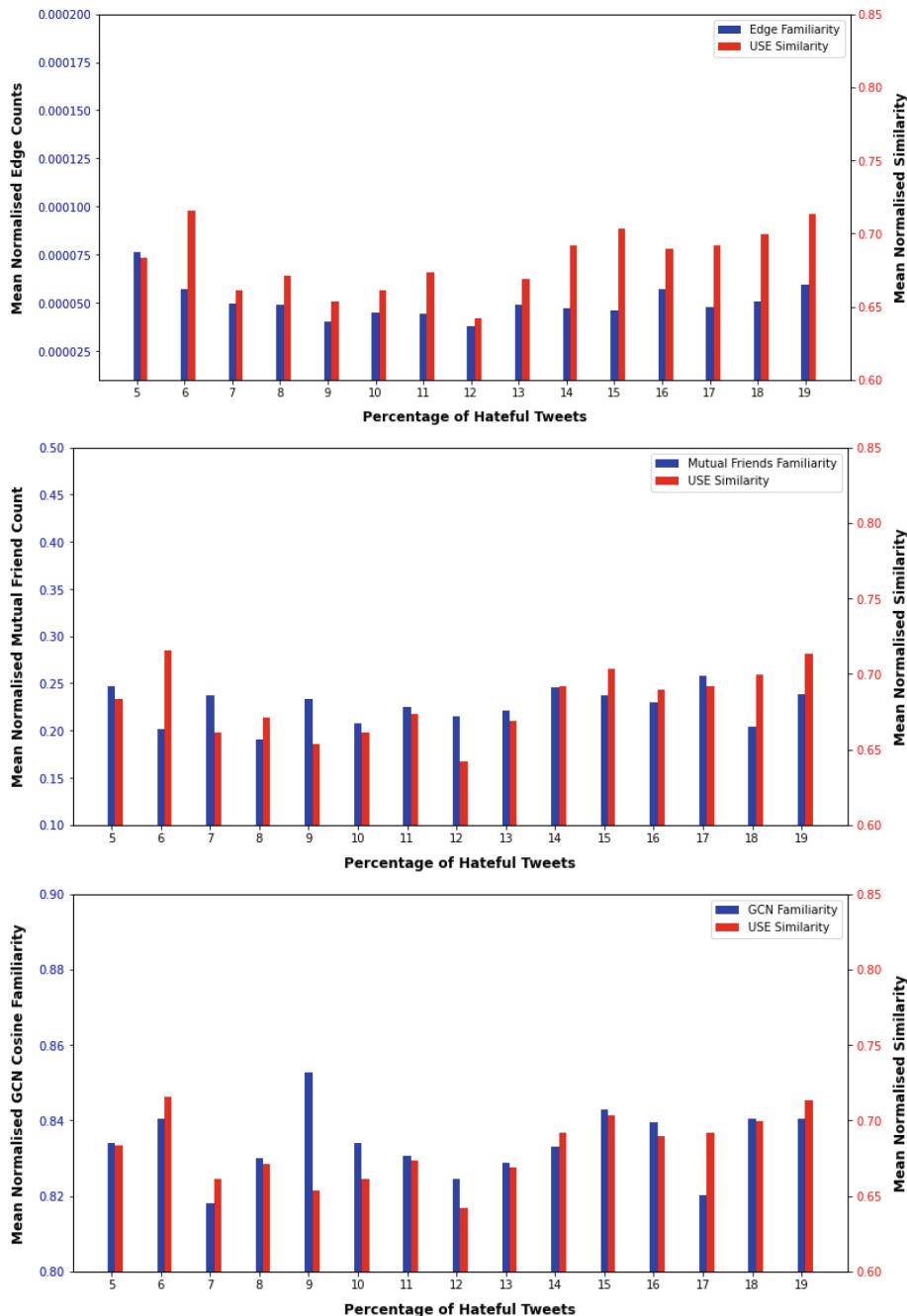


Fig. 2. Variation in similarity and familiarity as hatefulness increases

To answer RQ3, we create a user base for each hateful form (topic). We pick users whose affinity score is above a certain threshold. We also rank the different hashtags used by users by frequency. This is shown in Table 1. We observe that many users show higher values of association with specific topics (Topics - 0, 5, 7), as compared to the rest. Therefore, we decide to have a dynamic threshold for topic affinity. To compute these affinity thresholds, we select users in such a way that there are a reasonable number (at least 10% of total users) of representative users. For each topic, we plot the average familiarity, using cosine similarity of graph embeddings, and average similarity in Fig. 3. The similarity and familiarity values are normalized by dividing by the maximum values, respectively. We observe that topics 2, 5 and 7 exhibit stronger homophily, as compared to others for both the communities. These topics can be broadly categorized into hate manifesting as sexism, nationalism and racism.

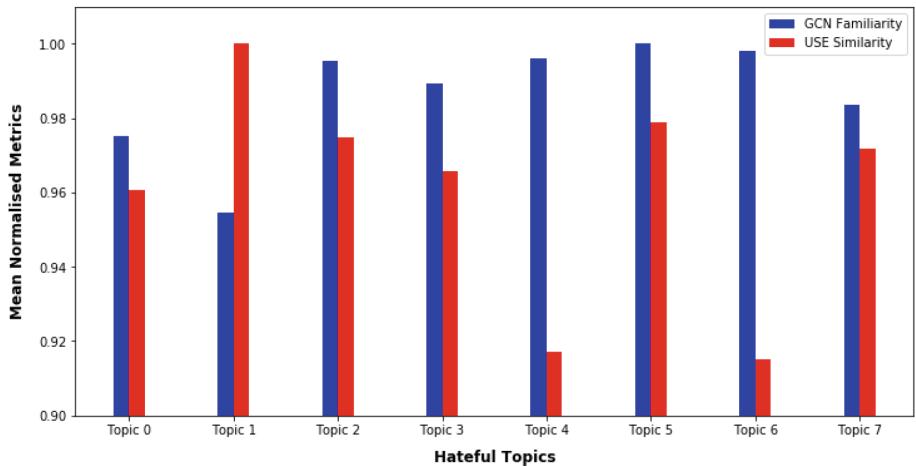


Fig. 3. Homophily for the different hateful forms

6 Conclusion

In this paper, we study homophily in hate speech on online social media platforms. We propose a novel way of computing familiarity using graph embeddings generated by variational graph auto-encoders. We empirically demonstrate that homophily is a significant driver in hate speech generation and also establish the effectiveness of graph embeddings in computing familiarity. Further, we observe the variation of homophily in different forms of hate. We report that there exists stronger homophily in users associated with certain hateful forms, as compared with others. These are racism, and xenophobia (nationalism). Therefore, we observe that homophily does play a role in the manifestation of hate speech. Further, we emphasize that the facet of homophily should be utilized for enhancing further research in this area.

References

1. Afrasiabi Rad, A., Benyoucef, M.: Similarity and ties in social networks a study of the YouTube social network. *J. Inf. Syst. Appl. Res.* **7**(4), 14 (2014)
2. Agrawal, R., Arquam, M., Singh, A.: Community detection in networks using graph embedding. *Procedia Comput. Sci.* **173**, 372–381 (2020)
3. Aral, S., Muchnik, L., Sundararajan, A.: Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proc. Natl. Acad. Sci.* **106**(51), 21544–21549 (2009)
4. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
5. Cai, H., Zheng, V.W., Chang, K.C.C.: A comprehensive survey of graph embedding: problems, techniques, and applications. *IEEE Trans. Knowl. Data Eng.* **30**(9), 1616–1637 (2018)
6. De Choudhury, M., Sundaram, H., John, A., Seligmann, D.D., Kelliher, A.: “Birds of a feather”: does user homophily impact information diffusion in social media? *arXiv preprint arXiv:1006.1702* (2010)
7. Dey, K., Shrivastava, R., Kaushik, S., Garg, K.: Assessing topical homophily on Twitter. In: Aiello, L., Cherifi, C., Cherifi, H., Lambiotte, R., Lió, P., Rocha, L. (eds.) *COMPLEX NETWORKS 2018. Studies in Computational Intelligence*, vol. 813, pp. 367–376. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-05414-4_29
8. Ducheneaut, N., Yee, N., Nickell, E., Moore, R.J.: The life and death of online gaming communities: a look at guilds in world of warcraft. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 839–848 (2007)
9. Halberstam, Y., Knight, B.: Homophily, group size, and the diffusion of political information in social networks: evidence from Twitter. *J. Public Econ.* **143**, 73–88 (2016)
10. Jeong, C., Jang, S., Shin, H., Park, E., Choi, S.: A context-aware citation recommendation model with bert and graph convolutional networks. *arXiv preprint arXiv:1903.06464* (2019)
11. Kipf, T.N., Welling, M.: Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016)
12. Mathew, B., Kumar, N., Goyal, P., Mukherjee, A., et al.: Analyzing the hate and counter speech accounts on Twitter. *arXiv preprint arXiv:1812.02712* (2018)
13. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: homophily in social networks. *Ann. Rev. Sociol.* **27**(1), 415–444 (2001)
14. Pan, S., Hu, R., Long, G., Jiang, J., Yao, L., Zhang, C.: Adversarially regularized graph autoencoder for graph embedding. *arXiv preprint arXiv:1802.04407* (2018)
15. Ribeiro, M., Calais, P., dos Santos, Y., Almeida, V., Meira Jr, W.: “Like sheep among wolves”: characterizing hateful users on Twitter (2017)
16. Röder, M., Both, A., Hinneburg, A.: Exploring the space of topic coherence measures. In: *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pp. 399–408 (2015)
17. Starbird, K., Palen, L.: (How) will the revolution be retweeted? Information diffusion and the 2011 Egyptian uprising. In: *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, pp. 7–16 (2012)
18. Weng, J., Lim, E.P., Jiang, J., He, Q.: TwitterRank: finding topic-sensitive influential Twitterers. In: *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, pp. 261–270 (2010)
19. Xu, S., Zhou, A.: Hashtag homophily in Twitter network: examining a controversial cause-related marketing campaign. *Comput. Hum. Behav.* **102**, 87–96 (2020)
20. Ying, J.J.C., Lu, E.H.C., Lee, W.C., Weng, T.C., Tseng, V.S.: Mining user similarity from semantic trajectories. In: *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, pp. 19–26 (2010)



Influencing the Influencers: Evaluating Person-to-Person Influence on Social Networks Using Granger Causality

Richard Kuzma^(✉), Iain J. Cruickshank, and Kathleen M. Carley

CASOS Institute, Carnegie Mellon University, Pittsburgh, USA
`{rkuzma2, icruicks}@andrew.cmu.edu,`
`kathleen.carley@cs.cmu.edu`
<http://www.casos.cs.cmu.edu/>

Abstract. We introduce a novel method for analyzing person-to-person content influence on Twitter. Using an Ego-Alter framework and Granger Causality, we examine President Donald Trump (the Ego) and the people he retweets (Alters) as a case study. We find that each Alter has a different scope of influence across multiple topics, different magnitude of influence on a given topic, and the magnitude of a single Alter’s influence can vary across topics. This work is novel in its focus on person-to-person influence and content-based influence. Its impact is two-fold: (1) identifying “canaries in the coal mine” who could be observed by misinformation researchers or platforms to identify misinformation narratives before super-influencers spread them to large audiences, and (2) enabling digital marketing targeted toward upstream Alters of super-influencers.

Keywords: Influence · Social media · Granger causality

1 Introduction

Influence in social networks remains one of the mainstays of social network analysis. Recently, with the advent of online social networks and the influence that online misinformation and disinformation can have, social influence has taken on added importance and dimensions. In this work, we build upon the foundations of social influence by examining social influence at the Ego network level. Using an Ego-Alter framework we identify an Ego (Donald Trump), examine which Twitter accounts he retweets most to identify key Alters (e.g. Ivanka

The research for this paper was supported in part by the Knight Foundation and the Office of Naval Research Grant (N000141812106) and an Omar N. Bradley Fellowship, and by the center for Informed Democracy and Social-cybersecurity (IDeAS) and the center for Computational Analysis of Social and Organizational Systems (CASOS) at Carnegie Mellon University. The views and conclusions are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Knight Foundation, Office of Naval Research or the US Government.

Trump, Lindsay Graham), identify tweet topics from the Ego using unsupervised machine learning, build a supervised machine learning classifier to classify tweets of Alters into those topics [2, 3]. Then for a given topic, we measured the influence of each Alter on the Ego by measuring the Granger causality on the Alter's tweet time series on the Ego's tweet time series.

We devised this method in order to answer the question, "who influences the influencers?" Given the role that social media super-influencers may play in the spread of (mis)information online, identifying who influences them, what topics they are influenced on, and how much they are influenced could be key to future misinformation studies (Fig. 1).

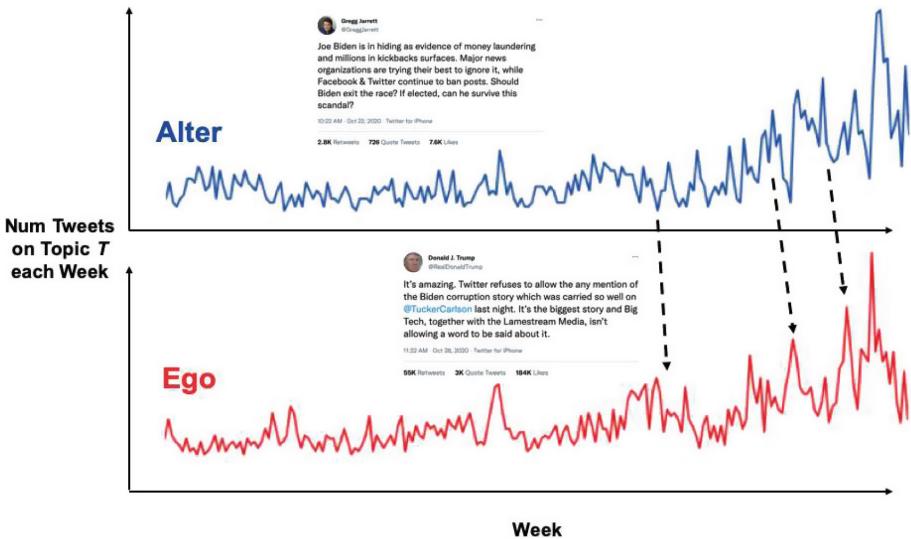


Fig. 1. Example of an Alter's Tweets affecting the Ego's Tweets for a given topic. Certain topics in the Alter's tweets are then Tweeted, with a time-delay by the Ego.

Our work is novel from other applications of Granger causality for a few reasons. Most Granger causality research on social media focuses on (1) population-level influence, (ours focuses on person-to-person influence); (2) sentiment analysis, (ours focuses on topics analysis); and (3) impact of social media behavior on real world events such as financial markets or protests (our work focuses on influence within online behavior on Twitter) [4, 7, 27]. Other work using Granger causality to examine person-to-person influence on social media examined sharing of explicit URLs between Twitter accounts, but not the content or topic of tweets as we do [35].

2 Background

The last few years have seen incredible advances in, and need for, social cybersecurity. Social Cybersecurity is an emerging field that looks at the intersection of human behavior in a cyber mediated information environment [10]. In particular, recent events like the COVID-19 pandemic and use of online disinformation in various national elections have highlighted the need to better understand the ways in which mis- and disinformation are spread online [8,36]. Previous works have generally focused on the role of bots and trolls in their roles in spreading mis- and disinformation [5,6,10,24,26,37]. More recently, coordination among cyborg or fully human accounts has been studied as means by which malicious actors can propagate mis- and disinformation [20,30,31]. The insight of coordinated spread by social media accounts is even being used to investigate cross-platform mis- and disinformation spread [34]. To date, much of the research in online mis- and disinformation campaigns has focused on particular types of accounts and coordination among accounts.

One area of the spread of online mis- and disinformation that is less well-studied is the role of online social media influencers. Social media influencers, or those online accounts with the ability to reach hundreds of thousands, if not millions, of other social media users with their content represent an important component of the social media landscape, by virtue of their reach. Recent work has shown that social media influencers can often have an outsized impact on the information quality of any given online discussion. For example, one study showed that the former U.S. President Donald Trump was responsible for over 30% of the COVID-19 misinformation on Twitter [15]. As another example, the ‘dirty dozen’, a collection of only twelve accounts, are responsible for the vast majority of anti-vaccination content online [14]. Additionally, recent disinformation campaigns, like Secondary Infektion have deliberately sought to target social media influencers to get them to spread disinformation [1,17]. Thus it is clear that social media influencers can play a tremendous role in the spread of mis- and disinformation.

It is also important to note that the study of influence in social networks has a long history with much research into the topic. The study of influence in social networks pre-dates online social networks and has been used to understand everything from individuals’ behaviors to collective action of groups [13,18,19]. Many of the models designed for understanding social influence work at the adjacency matrix-level of the social networks, and are fundamentally matrix based models [16,28]. As such, these models tend to look more at population levels of influence. Less studied within the social network analysis community is the role and exercise of social influence of Ego networks.

An important consideration when studying influence is the role of causality. Granger causality is a statistical method that examines whether X has predictive power in describing Y [22]. X_t is said to Granger-cause Y_t if Y_{t+1} can be better explained by past values of Y_t and past values of X_t than by the presence of past values of Y_t alone.

2.1 Data and Definitions

We retrieved tweets from Donald Trump from the Trump Twitter Archive [9]. This repository contains 56,571 tweets and retweets from Donald Trump's twitter account. Reports indicate that not all tweets from the @RealDonaldTrump twitter account come directly from Donald Trump. Prior to the presidency, it was believed that tweets from an Android device were from Donald Trump while other platforms (iPhone, web) were not. Once Donald Trump became president, he switched to a specially-issued iPhone. At least one staffer was trusted with @RealDonaldTrump twitter privileges. Tweets came from Donald Trump, direct dictations to staffers, and presumably from the close, trusted staffers themselves attempting to channel Donald Trump's head space [12]. Given the importance of the Twitter audience to Donald Trump, we believe that all tweets from the account, even those from staffers instead of Trump himself, accurately reflect Donald Trump's wishes and are worth studying as if they came from Trump himself (Fig. 2).

3 Methods

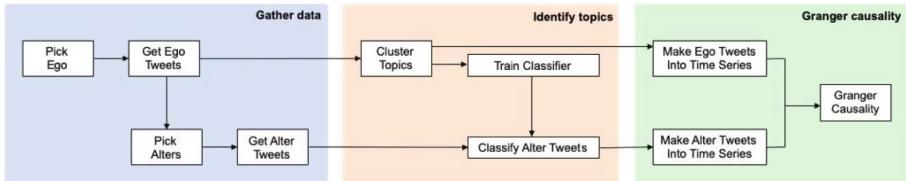


Fig. 2. Overview of Method. Our method consists of three, main steps: gathering the data and constructing the ego-alter network, finding topics in the textual data and building a classifier of those topics, and then testing for causality in the ego-alter network on those topics.

Our method has three stages: gathering data, identifying topics of conversation, and determining granger causality.

3.1 Gathering Data

We gathered tweets from the Trump Twitter Archive, which retained copies of all of Trump's tweets and retweets from before Trump's Twitter suspension. CSV and JSON files with all captured tweets are available at the Trump Twitter Archive. We limited tweets to the time that Donald Trump was President of the United States and active on Twitter, January 20th, 2017 to January 8th, 2020. We hypothesized that the information Trump ingested affected the information he output. In other words, what he retweeted affected what he tweeted.

We found 12 accounts from individuals Trump retweeted the most (the Alters). We excluded Trump’s retweets of @WhiteHouse and of himself (@RealDonaldTrump). These Alters included family members, conservative media pundits, and members of Congress. We used SNScrape, an open-source tool, to get the tweet IDs of these 12 Alter accounts [25]. The original tweets of these accounts were available, but data was not available on who these Alters retweeted. We used these tweet IDs to get the tweet content and metadata through a process known as hydration, using Hydrator, another open-source tool [29].

3.2 Identifying Topics

We cleaned Trump’s tweets by extracting URLs, mentions, and “RT” (used for manual retweets when Trump quoted another twitter user) from tweets. We also used the gensim Python library to preprocess tweets by removing punctuation and making all words lowercase. We then tokenized the tweets and found common bigrams and trigrams using the gensim library. Bigrams are two-word sequences like “crooked hillary” or “sleepy joe” and trigrams are three-word sequences like “make america great.” Following tokenization, we embedded the tweets using the Universal Sentence Encoder (USE) from Google Research which was available at the TensorFlow Hub [11, 21]. We also featurized all tweets using the term frequency-inverse document frequency (TF-IDF) from the scikit learn library [32].

We chose the USE because it was built for short sequences ranging from a sentence to a short paragraph in length, which is similar to the length of a tweet, and performs well in semantic similarity tasks, determining if one sequence is similar to another. We performed k-medoids clustering [38] on the USE embeddings to identify topic clusters and within those topic groups examined the most important words by averaging the TF-IDF scores of all documents within a given USE cluster.

We chose to cluster only on the Ego’s tweets because including Alter tweets in the topic classifier would dilute the Ego’s tweet content with Alter content when forming clusters. With 12 Alters in our case study, it is possible topics would drift toward Alter content rather than Ego content, making a classifier predict whether a given Alter’s tweet was like other Alter tweets rather than similar to the Ego’s tweet. We settled on eight topics for Trump’s tweets through a combination of silhouette score [33] from the k-medoid clustering of USE embeddings and interpretation of the TF-IDF top words for each cluster (Fig. 3).

We categorized each Alter’s tweets into one of the categories of Trump’s tweets. To do this we train a Support Vector Classifier on Trump’s tweets, using the USE embeddings as features and the unsupervised topic clusters as the label. Our final model had a weighted F-1 score of 0.91.

For each Alter, we clean the tweets using the same cleaning pipeline as Trump’s tweets then embed the tweets using USE embeddings. We then use the model trained to predict Trump’s tweet categories to predict the Alter’s tweet categories. Now that the Alter’s tweets are categorized, we turn them into eight time series, one for each of the topic clusters. We use weeks as time-steps

Topic	Top 50 words
Fake News	fake_news, report, media, fake_news_media, great, cnn, witch_hunt, watch, russia_story, bad, say, news, democrat, people, fbi, time, trump, know, corrupt, mueller, collusion, big, lie, today, look, fake, dems, election, history, thank, mueller_report, investigation, want, make, ukraine, fox_news, fact, poll, cover, rating, good, try, lamestream_media, country, like, use, fox, new, real
Congress	thank, democrat, vote, senate, dems, people, nancy_pelosi, great, republican, want, senator, american_people, time, impeachment, year, joe_biden, work, know, try, wow, country, say, need, congress, like, watch, house, nothing_democrat, think, biden, pelosi, let, right, happy_birthday, way, good, happen, make, big, chuck_schumer, pass, adam_schiff, remember, american, lose, thing, help, come, care, fight
MAGA	great, big, vote, country, china, economy, new, job, thank, people, today, make, year, want, tariff, good, win, come, deal, congratulation, work, farmer, trade, agree, best, time, democrat, strong, help, mexico, world, fight, need, way, number, look, history, pay, high, day, trade_deal, united_state, state, american, america, company, president, love, like, usa
Democrat Obstruction	democrat, impeachment, republican, trump, job, dems, want, vote, president, impeach, say, election, time, law_order, day, presidential_harassment, approval_rating_republican_party_thank, reason, president_trump, think, congress, house, thank, try, republican_party, today, make, year, big, win, justice, know, like, great, watch, need, attack, daca, people, way, work, political, case, antifa, police, crime, house Democrat, right, break, fast
Campaigning	great, state, thank, people, true, vote, pennsylvania, governor, fake_news, election, country, work, make, know, interview, today, say, president, like, job, joe_biden, live_press_brief_coronavirus_task_force, watch, north_carolina, want, fact, great_job, big, voter_fraud, ballot, allow, joe, fraud, mayor, let, good, stop, time, way, book, report, right, sleepy_joe, biden, use, story, voter, test, case, new_york
America First	president, great, people, country, democrat, support, military, help, big, american, thank, america, strong, want, day, need, good, united_state, vote, today, protect, bad, border, know, biden, world, safe, nation, republican, year, border_security, job, iran, state, fight, joe_biden, important, way, stand, win, time, make, come, leader, history, economy, veteran, work, china, peace
Praising Trump	thank, trump, president, president_trump, great, make_america_great, america, win, today, love, american, congratulation, country, donald_trump, big, vote, day, florida, way, united_state, job, people, white_house, biden, work, work_hard, proud, incredible, great_job, year, make, honor, tonight, great_honor, live, election, time, support, rally, million, best, come, welcome, like, help, happy, wisconsin, usa, melania, sign
Investigations	president, whistleblower, fbi, schiff, say, impeachment, adam_schiff, make, know, witch_hunt, democrat, case, people, ask, investigation, witness, want, lie, report, congress, evidence, release, hoax, comey, use, come, order, day, american, break, new, testify, start, think, obama, happen, hear, investigate, leak, like, time, fact, impeachment_hoax, ukraine, tell, bad, country, email, story, lawyer

Fig. 3. Top 50 words and phrases used by Donald Trump in each topic (ranked by TF-IDF score). Clusters determined by USE embeddings.

because we found days contained too few tweets to make an effective time series for each topic.

3.3 Granger Causality

We create a time-series with units of weeks on the x-axis and number of user's tweets on a given topic on the y-axis. The Ego, E , and Alters, A_1 to A_n , each each have m time series, where m is the number of topics. Then, for a given Alter (e.g. Ivanka Trump) and a given topic (e.g. Campaigns) we want to find if the Alter's tweets granger-cause (have predictive power to describe) the tweets of the Ego (e.g. Donald Trump).

We first considered granger causality at the level of days, but found that most users did not have enough data when viewing topics at the day level. We think weeks are a viable time scale because intuition suggests that an Alter may tweet about a topic of importance to them repeatedly. In future work, it would be worth considering a 2–3 day rolling window.

Testing for granger causality requires that both time series are stationary, that is, the mean, variance, and autocorrelation do not change with time. Stationary was checked with the Augmented Dickey Fuller test [23]. If a time series

is not stationary, it may be made stationary by differencing the time series. This requires subtracting the number of tweets at time t from the number of tweets at time $t - 1$. If the time series of the Alter's or Trump's tweets for a given topic are not stationary, both time series were differenced and stationarity re-examined. This way both time series always had the same amount of differencing.

If the time series could be made stationary, a granger causality test can then be done. Again, this test determines if Y_t (Trump's tweets on a topic) is better explained by Y_{t-n} (Trump's previous tweets on that topic) alone or Y_{t-n} and X_{t-n} (Alter's previous tweets on that same topic).

4 Results

Different Alters influence the Ego on different topics. Even among Alters whose tweets granger-cause Trump's tweets, some Alters have a more immediate effect (1 week) while others affect Trump's tweets with six to eight weeks of lag.

Members of Congress (Mark Meadows, Jim Jordan, GOP Chairwoman) had influence on the “Congress” topic with one week lag between their tweets and Trump's. Jim Jordan and GOP Chairwoman also influenced the “Investigations” topic with a one week lag.

Conservative pundits (Mark Levin, Gregg Jarrett, and Tom Fitton) influenced the most topics (6–7), though not all with immediate effect. Their tweets on “Fake News,” “Campaigning,”, and “Investigations” (along with pundit Charlie Kirk, though less so with Tom Fitton) had 1–3 week lags between their tweets and Trumps.

Donald Trump Jr had influence on “Fake News,” “Congress,” “Campaigning,” and “America First” topics, all with 1–3 week lags. Ivanka Trump only influenced “America First” and “Praising Trump” but with 5–6 week lags. Eric Trump had influence on more topics, but a mix of immediate lag (1–3 week) and delayed lag (6–7 week) influence.

Among Alters that influenced five or more topics (Eric Trump, Mark LEvin, Gregg Jarrett, and Tom Fitton), the immediacy of influence was not consistent. Often the Alter had a few topics with immediate lag influence (1–2 week) and a few with delayed lag influence (7–8 week).

We chose to do individual hypothesis testing and felt the low p-values would not substantively change with multiple hypothesis testing, but future work may choose to multiple hypothesis testing for added rigor (Fig. 4).

5 Discussion

In this paper we ask “who influences the influencers?” and introduce a novel method for examining person-to-person influence on Twitter using an Ego-Alter framework for finding which Alters influence the Ego's posts on social media, on which topics these Alters influence the Ego, and the time delay between Alter posts and Ego posts. We show this specifically with tweets from Donald Trump and the first-order Alters he interacts with on Twitter, but believe this work

Alter	Fake News	Congress	MAGA	Democrat Obstruction	Campaigning	America First	Praising Trump	Investigations	Avg Lag /Alter
Eric Trump	-	-	0.01 (1)	0.008 (7)	< 0.001 (3)	0.016 (1)	0.027 (6)	-	3.6
Ivanka Trump	-	-	-	-	-	0.021 (5)	0.026 (6)	-	5.5
Donald Trump Jr	0.05 (1)	0.07 (1)	-	-	0.028 (3)	0.001 (1)	-	-	1.5
Dan Scavino	-	-	-	-	-	-	-	-	-
Mark Meadows	-	0.004 (1)	-	-	-	-	-	-	1
Jim Jordan	-	0.001 (1)	-	-	0.003 (2)	-	-	0.004 (1)	1.3
Lindsay Graham	-	-	-	-	-	-	-	-	-
GOP Chairwoman	-	< 0.001 (1)	-	-	-	0.043 (4)	0.036 (4)	0.007 (1)	2.5
Mark Levin	0.031 (3)	0.044 (7)	0.002 (2)	0.038 (7)	< 0.001 (1)	-	-	0.031 (3)	3.83
Gregg Jarrett	0.047 (1)	0.003 (2)	< 0.001 (5)	-	0.012 (3)	0.017 (7)	0.029 (2)	< 0.001 (1)	3
Charlie Kirk	-	-	-	0.039 (1)	-	-	-	< 0.001 (1)	1
Tom Fitton	0.005 (1)	0.023 (6)	0.027 (8)	-	-	0.002 (1)	0.038 (3)	0.017 (6)	4.17
Avg Lag /Topic	1.5	2.7	4.5	5	2.4	3.17	4.2	2.17	

Fig. 4. P-Values for granger causality test. Num lags in parenthesis (max 8 weeks). Alters shown in three groups: (1) family/close advisors, (2) members of Congress, and (3) political commentators.

could be extended to examine other influencers on Twitter and their networks, to include second and third-order Alters farther away from the Ego's immediate network, and attempted on other social media platforms (Parler, Facebook).

Our results show substantial variation in the influence of Alters on an Ego. While our work confirms the general theories of social influence in that social influence affects an individual's behavior, we also note what influences an individual and when they influence that individual are heterogeneous entities. Some Alters have a greater scope of influence. Of Donald Trump's eight tweet topics, some Alters influence only one or two topics while others influence seven of the eight topics. Some Alters have narrow, but powerful, influence. Charlie Kirk, a conservative talk show host, influences Trump on only two topics, "investigations" and "democrat obstruction." Trump's tweets lag Kirk's by only one week, showing that Kirk has high influence on these topics. Five other Alters influence the "investigations" topic and three of them also have a one-week lag on their influence, so Kirk is not special in this category. On "democrat obstruction," however, only two other Alters with influence. Kirk has a one-week lag between his "democrat obstruction" tweets and Trumps, while the other Alters have a seven-week lag, indicating less influence on this topic.

Different Alters influence different topics and should not be viewed as substitutable. Tom Fitton, a conservative activist and President of Judicial Watch, is very influential in the "fake news," "america first," and "praising trump" categories whereas Jim Jordan, a Congressman from Ohio, is highly influential on topics of "congress," "campaigning," and "investigations."

Some Alters influence different topics differently. This suggests that some Alters are more influential on certain topics than others. For instance, Gregg Jar-

rett, a conservative American television commentator, influences Trump’s tweets on seven of eight topics. On “fake news” and “investigations” his tweets influence Trump’s after a one-week lag. On “MAGA” and “America First” his tweets influence Trump’s after a five-week and seven-week lag, respectively. Jarett would be a good influence vector for “fake news” and “investigations” but not as good for “MAGA” and “America First”

5.1 Future Work

Despite the useful results and demonstration of using granger-causality in understanding peer-to-peer influence on social media, there are limitations to this work. We examine a specific case of influence surrounding Donald Trump, without any comparison to how these Alters affected other Twitter users or how other Egos are influenced by their respective Alters. We focused on this base case because the Trump Twitter Archive permitted us to study the retweets of the Ego to identify the most-retweeted Alters. Generally, retweets of specific Twitter accounts are not available through the Twitter API nor are they available through Python libraries like SNScrape, making data collection difficult. If retweets of Alters were available, research could be done on whether second-order Alters (e.g. someone who Ivanka Trump retweets a lot) has influence on the Ego (Donald Trump) through the first-order Alter (Ivanka Trump).

We see two uses of this influence work. First, in the disinformation space, using Alters as “canaries in the coal mine” who could be observed by misinformation researchers or platforms to notice new misinformation narratives before super-influencers catch hold of these narratives and spread them to millions of people. Second, in the marketing space, advertisers may not have the budget to pay a super-influencer to market their product, but by paying the Alters of the Ego, the advertiser may be able to influence the super-influencer to support the product being marketed.

References

1. Secondary infektion (2021). <https://secondaryinfektion.org/>
2. Arnaboldi, V., Conti, M., Passarella, A., Pezzoni, F.: Analysis of ego network structure in online social networks. In: International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing, vol. 10, pp. 31–40 (2012)
3. Arnaboldi, V., Conti, M., Passarella, A., Pezzoni, F.: Ego networks in Twitter, an experimental analysis, pp. 3459–3464 (2013)
4. Bastos, M.T., Mercea, D., Charpentier, A.: Tents, tweets, and events: the interplay between ongoing protests and social media. *J. Commun.* **65**(2), 320–350 (2015)
5. Beskow, D.: Finding and characterizing information warfare campaigns. Ph.D. thesis (2020)
6. Bessi, A., Ferrara, E.: Social bots distort the 2016 U.S. presidential election online discussion. *First Monday* **21**(11) (2016)
7. Bollen, J., Mao, H., Zeng, X.: Twitter mood predicts the stock market. *J. Comput. Sci.* **2**(1), 1–8 (2011)

8. Brangham, W.: “Tech heads testify on misinformation in the aftermath of Jan 6 riots.” PBS. org. <https://www.pbs.org/newshour/show/tech-heads-testify-on-misinformation-in-the-aftermath-of-jan-6-riots>. Accessed 25 Mar 2021
9. Brendan: Trump Twitter archive v2 (2021). <https://www.thetrumparchive.com/>
10. Carley, K.M.: Social cybersecurity: an emerging science (2020)
11. Cer, D., et al.: Universal sentence encoder. arXiv (2018). <https://arxiv.org/abs/1803.11175>
12. Chute, N.: He writes trump’s tweets and has been with trump’s campaign since day one (2019)
13. Coussi-Korbel, S., Fragaszy, D.M.: On the relation between social dynamics and social learning. *Anim. Behav.* **50**(6), 1441–1453 (1995)
14. Dozen, T.D.: Center for Countering Digital Hate (2021)
15. Evanega, S., et al.: Quantifying sources and themes in the covid-19 ‘infodemic’ (2020)
16. Flache, A., Mäs, M., Feliciani, T., Chattoe-Brown, E., Deffuant, G., Huet, S., Lorenz: models of social influence: towards the next frontiers. *J. Artif. Soc. Soc. Simul.* **20**(4) (2017). <https://doi.org/10.18564/jasss.3521>, <http://jasss.soc.surrey.ac.uk/20/4/2.html>
17. Metamorphosis Foundation: Disinformation about Belarus spreads in the Balkans via online portals and social media (2020)
18. Friedkin, N., Johnsen, E.: Social influence network theory: a sociological examination of small group dynamics (structural analysis in the social sciences) (2011)
19. Friedkin, N.E., Proskurnikov, A.V., Tempo, R., Parsegov, S.E.: Network science on belief system dynamics under logic constraints. *Science* **354**(6310), 321–326 (2016). <https://science.sciencemag.org/content/354/6310/321>
20. Giglietto, F., Righetti, N., Rossi, L., Marino, G.: It takes a village to manipulate the media: coordinated link sharing behavior during 2018 and 2019 Italian elections. *Inf. Commun. Soc.* **23**(6), 867–891 (2020)
21. Google: Universal sentence encoder (computer software) (2018)
22. Granger, C.W.J.: Investigating causal relations by econometric models and cross-spectral methods. *Econometrica* **37**(3), 424–438 (1969)
23. Greene, W.H.: Econometric Analysis. Macmillan Publishing Company (199)
24. Hui, P.M., et al.: BotSlayer: real-time detection of bot amplification on Twitter. *J. Open Source Softw.* **4**(42), 1706 (2019). <https://doi.org/10.21105/joss.01706>
25. JustAnotherArchivist: snscreape (computer software) (2021). <https://github.com/JustAnotherArchivist/snscreape>
26. McKenzie, H.W., et al.: Bots and misinformation spread on social media: implications for COVID-19. *J. Med. Internet Res.* **23**(5), e26933 (2021)
27. Mittal, A., Goel, A.: Stock prediction using Twitter sentiment analysis (2011). <http://cs229.stanford.edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis.pdf>
28. Nedić, A., Olshevsky, A., Uribe, C.A.: Graph-theoretic analysis of belief system dynamics under logic constraints. *Sci. Rep.* **9** (2019)
29. the Now, D.: Hydrator (computer software). <https://github.com/docnow/hydrator>
30. Pacheco, D., Hui, P., Torres-Lugo, C., Truong, T., Flammini, A., Menczer, F.: Uncovering coordinated networks on social media: methods and case studies (2020)
31. Pacheco, D., Flammini, A., Menczer, F.: Unveiling coordinated groups behind white helmets disinformation. In: Companion Proceedings of the Web Conference 2020 (WWW 2020). Association for Computing Machinery, pp. 611–616 (2020)

32. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
33. Shahapure, K.R., Nicholas, C.: Cluster quality analysis using silhouette score. In: 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), pp. 747–748 (2020)
34. Starbird, K., Wilson, T.: Cross-platform disinformation campaigns: Lessons learned and next steps (2020)
35. Steeg, G., Galstyan, A.: Information transfer in social media. In: Proceedings of the 21st International Conference on World Wide Web, pp. 509–518 (2012)
36. Taylor, N., Nicole, K., Claire, C., Alan, H., Albert, H.: The danger of misinformation in the COVID-19 crisis. *Mo. Med.* **117**(6), 510–512 (2020)
37. Uyheng, J., Carley, K.M.: Characterizing network dynamics of online hate communities around the COVID-19 pandemic. *Appl. Netw. Sci.* **6**(20), 1–21 (2021)
38. Velmurugan, T., et al.: Efficiency of k-means and k-medoids algorithms for clustering arbitrary data points. *Int. J. Comput. Technol. Appl.* **3**(5), 1758–1764 (2012)



How the Far-Right Polarises Twitter: ‘Hashjacking’ as a Disinformation Strategy in Times of COVID-19

Philipp Darius¹ and Fabian Stephany^{2,3(✉)}

¹ Centre for Digital Governance, Hertie School, Berlin, Germany

² Oxford Internet Institute, University of Oxford, Oxford, UK
fabian.stephany@oii.ox.ac.uk

³ Humboldt Institute for Internet and Society, Berlin, Germany

Abstract. Twitter influences political debates. Phenomena like *fake news* and *hate speech* show that political discourses on social platforms can become strongly polarised by algorithmic enforcement of selective perception. Some political actors actively employ strategies to facilitate polarisation on Twitter, as past contributions show, via strategies of ‘hashjacking’ (The use of someone else’s hashtag in order to promote one’s own social media agenda.). For the example of COVID-19 related hashtags and their retweet networks, we examine the case of partisan accounts of the German far-right party *Alternative für Deutschland* (AfD) and their potential use of ‘hashjacking’ in May 2020. Our findings indicate that polarisation of political party hashtags has not changed significantly in the last two years. We see that right-wing partisans are actively and effectively polarising the discourse by ‘hashjacking’ COVID-19 related hashtags, like #CoronaVirusDE or #FlattenTheCurve. This polarisation strategy is dominated by the activity of a limited set of heavy users. The results underline the necessity to understand the dynamics of discourse polarisation, as an active political communication strategy of the far-right, by only a handful of very active accounts.

Keywords: COVID-19 · Hashtags · Networks · Political communication strategies · #CoronaVirusDE · #FlattenTheCurve

1 Introduction

The Coronavirus pandemic was accompanied by a so-called ‘infodemic’. This term refers to the acceleration of information and high uncertainty in the information environment, as for instance, in science, the media or policymaking. Moreover, there was a surge of dis- and misinformation and most debated ‘conspiracy narratives’ during the pandemic that additionally disrupted the media and information system. In this article, we show that far-right actors used long established strategies such as ‘hashjacking’ - using political opponents hashtags and public hashtags - to influence public opinion formation and leverage their

own content. Consequently, we assess polarisation strategies by partisans of the German right-wing populist party “Alternative für Deutschland” during two observation periods in 2018 and 2020. The analytical approach builds on community detection algorithms from social network analysis and logistic regression models to determine the likelihood of strategic hashtag hijacking or “hashjacking”.

The results indicate that far-right users and politicians have been successful in establishing counterpublics on Twitter and were much more involved in linking their messages to wider discourses than partisans of other parties. By comparing the polarisation of party hashtags between 2018 and 2020, we identify political partisan groups and show that polarisation of party hashtags is relatively stable and highest for the far-right #AFD. Moreover, our results indicate that Twitter debates on #CoronaVirusDE and #FlattenTheCurve are strongly polarised, too. The group of accounts spreading sceptical messages, questioning the existence of COVID-19 or the necessity of measures taken against the virus, is populated by partisans of the far-right. Within the far-right partisan movement around #AFD a handful of heavy users are retweeting most of the content. It is important to understand the dynamics of social media discourse, when engaging in it: A small group of political partisans, in our case of the far-right AfD, systematically polarises discussions on public hashtags to influence public opinion formation and leverage their own content.

2 Background

Social platforms have become central hubs and fora for political debates, economic transactions, marketing and social communication. As a consequence, these platforms and the companies that own the infrastructure take an increasingly important role in the formation of public opinion, political campaigning and political news consumption. They act as gatekeepers of information or “custodians of the Internet” [1]. In recent years, however, phenomena like political polarisation, radicalisation, and extremism are perceived notably in relation to online harms such as disinformation and abusive speech on these platforms. For extreme parties and their political narratives, social media offer additional channels for political communication in which extreme political actors do not need to follow the values and norms of traditional mass media and are thus able to spread their respective ideology [2]. While the ideologies of far-right parties are often called “thin”, their essence can be summarised as the rhetoric construction of (1) anti-elitism/establishment, (2) anti-migration, (3) anti-Muslim stances. Notably these three pillars of right-wing populist rhetoric and policies have in common to be against something and in particular to be against certain groups of people [3].

Social platforms and messengers are particularly attractive for politicians from right-wing populist and radical-right parties who often have a hostile attitude towards established media and sometimes limited access to traditional media channels [2]. Thus, right-wing populist actors and movements benefited disproportionately from the emergence of social media, since they could bypass

traditional media gatekeepers and communicate directly with their respective target audiences [4]. Besides, the direct contact to political actors and the represented ideologies enables the effect of a self-socialisation of citizens and users into right-wing populist beliefs and worldview [5]. Furthermore, social media also provide an opportunity for top-down claims of leadership for populist parties and politicians [5]. What remains unclear though are the logic and mechanisms that underlie these communication strategies.

To better understand the anti-pluralist ideology and mechanisms of strategic communication by far-right actors on social media, we refer to one of the most controversial political theorist of the 20th century, the German law professor Carl Schmitt, who later had close ties to the Nazi regime. His theory on the state of exemption (“Ausnahmezustand”) is an often-used term and was proclaimed as a tool for power by the Nazis a couple of years after his writing on the subject [6,7]. But what we find most resembling of current online and offline phenomena is the formulation of the political as the identification of individuals to one specific group in an enemy-friend scheme that in his view characterises any fundamental political conflict and varies by the perceived importance of the issue. Schmitt’s definition of the political argues against the ideas of consensus in liberal philosophy and constitutionalism as one of the central mechanisms in pluralist democracies. Individuals have to decide for either side and act as partisans in highly intensified or ‘existential’ political conflicts. This notion aligns with current rhetoric of right-wing populist parties in many countries and especially with the strategies and communication of alt-right actors online. Moreover, the political situation during Coronavirus pandemic resembles a phase of high intensity in political conflict since many people face uncertainty regarding their expectations towards the future on an individual and a societal level.

While an intensification of political groups could be observed by demonstrations of corona-sceptics on the streets of many capitals we also find this kind of friend-enemy polarisation in many online spaces where actors with radical beliefs form counterpublics and aim at influencing the political discourses and public opinion. While there may be various examples we find the following particularly representative of the Schmittian logic of the political. On social platforms political groups are in conflict, for instance, regarding the power of interpretation of popular political hashtags on Twitter. More or less organised groups attempt to ‘hijack’ hashtags, in order to change the meaning of the underlying concept or campaign. Thus, we will focus on these ‘hashjacking’ strategies as a tool of strategic political communication that also includes injecting disinformation into the mainstream/civil society discourse.

Social media have become an important political forum for the formation of political opinion and consequently also political persuasion and debate. When mapping these debates it is possible to identify different groups that arise due to ideologically aligned messaging behaviour [8,9]. One strategy used by the aforementioned more or less organised political groups that is especially relevant for Twitter is the hijacking or ‘hashjacking’ of hashtags [10–14]. This notion refers to the function of a hashtag to represent an issue, movement or organisation and

the general expectation that the stream of information and messages pertaining to these hashtags resembles in some way the public opinion on that matter [15–17]. Consequently, the action of ‘hashjacking’ as using hashtags that were established by politically opposed groups or a general civil society discourse, is executed with the goal of quantitatively dominating the content referring to this specific hashtag [14]. With regards to public hashtag debates, users using this strategy form an adverse group or counter-public that is separated from the wider discourse and are located in a separated cluster when illustrated in a network graph [18]. Consequently, the prior evidence in the literature informs the formulation of our hypotheses and the methodological approach which are introduced in the following sections.

3 Hypotheses

Revisiting our investigations on ‘hashjacking’ in 2018 [14] and based on the presented literature on ‘hashjacking’ strategies, we formulate the following hypothesis.

As the share of citizens for which the Internet and social platforms represent the most important channels for political news consumption is increasing and political discourse is increasingly taking place on platforms like Twitter, we assume that debates around the major political parties in Germany remain polarised.

(H1): The polarisation of retweet networks of political party hashtags did not change between 2018 and 2020.

Likewise, we assume that civil society discourse on Twitter relating to issues like the containment of the COVID-19 pandemic is subject to polarisation.

(H2): Retweet networks of COVID-19 related hashtags show a significant degree of polarisation.

As alternative communication channels (social platforms and messengers) are particularly attractive for right-wing populist and radical-right politicians, we assume that COVID-19 related hashtags, like #CoronaVirusDE and #FlattenTheCurve, have been systematically targeted by German far-right partisans of #AFD.

(H3): There is a significant activity of right-wing partisan accounts in contra-clusters of COVID-19 retweet networks.

4 Research Design

The study is based on Twitter data that was collected by accessing Twitter’s Rest API and using political party hashtags of German parties represented in the federal parliament (#AFD, #CDU, #CSU, #FDP, #GRUENE, #LINKE,

#SPD) and COVID-19 related hashtags (#CoronaVirusDE and #FlattenTheCurve) as a macro-level selection criterion. In total this study builds on a sample ($n=101,765$) of all public accounts using one or multiple of the selected political party hashtags between May 28th and June 4th in 2018 and 2020 on Twitter. The analysis focuses on a network approach and a visualisation of the retweet networks in Gephi using the Force2 layout algorithm [19] for each hashtag where retweeting creates a link (edge) between two accounts (nodes). Since the data was collected as separate streams of data pertaining to each hashtag, an account using several hashtags during the observation period in a retweet or being retweeted, will appear in each of the respective hashtag networks.

Political discourses on Twitter show polarised or clustered structures due to the retweeting behaviour, as indicated by the literature. The analysis, consequently, focuses on the retweeting networks of the chosen hashtags. In a first step of analysis the modularity (community detection) algorithm [20] assigns the nodes to different communities based on the structural properties of the network graph. The cluster membership is indicated by the colour of nodes as shown for #AfD in Fig. 4 in the Appendix. Thereafter, the interpretability of the clustering, as being in support of or opposition to a party, is controlled with a qualitative content analysis of the 50 most retweeted accounts [21]. This pro-/contra-polarisation of each party retweet network gives an indication of groups and whether individual nodes from these clusters are likely cluster of other party hashtags. Similarly, we perform a qualitative sentiment analysis of 30 randomly selected retweets (see Table 1).

Based on previous investigations [14], we know that a high pro-party X & contra-party Y association indicates ‘hashjacking’ strategies. Consequently, we use a logistic regression model to test all cluster combinations (as the likelihood to be in a contra-cluster of hashtag Y given a node was in the partisan cluster of party X). We decide to apply a logistic model for the assessment of cross-cluster heterophily, since our dependent variable is binary (location in contra-cluster) and the resulting odds are easy to interpret. Assuming there is no group that uses other party hashtags more frequently, users from all clusters should have the same odds to appear in the other network clusters. Thus, a high affiliation between two clusters in terms of their users being more likely to appear in both of them is an indication for strategic hashtag use or ‘hashjacking’.

5 Results

In the first part of the analysis, we compare the polarisation of German political party hashtags, i.e., the share tweets in either a pro and contra-cluster, between our data sample collected 2018 and 2020. As illustrated in Fig. 1, we observe that the share of polarised tweets did not change much between 2018 and 2020, thus confirming research hypothesis (1). Polarisation, in our definition, is still among highest for #AFD, while the polarisation, i.e., growth of the contra-cluster, has likewise increased for the hashtag of the Socialdemocrats (#SPD).

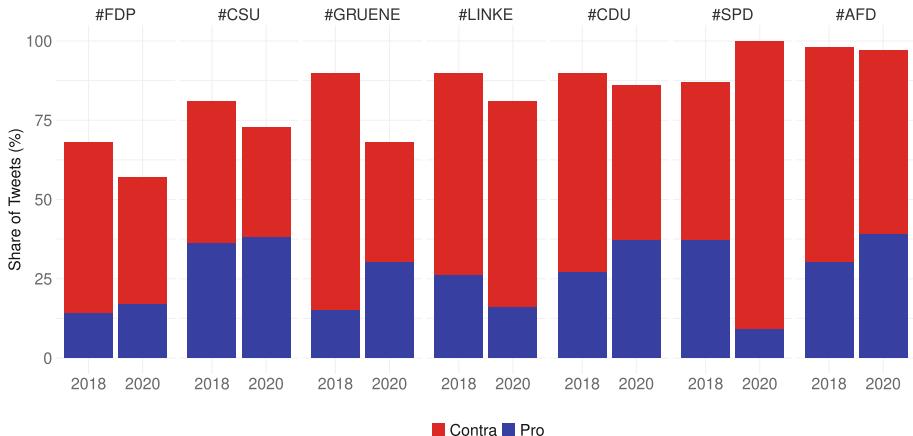


Fig. 1. In comparison to our data collected in May 2018, polarisation (the share of pro versus contra tweets) of the seven major German party hashtags has not changed significantly. It is still among highest for #AFD, while the polarisation (growth of the contra-cluster) has likewise increased for the hashtag of the Socialdemocrats (#SPD)

Secondly, similar to party hashtags, we construct retweeting networks for two COVID-19 related hashtags most popular in Germany in May 2020: #CoronaVirusDE and #FlattenTheCurve. With the example of #CoronaVirusDE, Fig. 2 illustrates the shape of these networks. After the application of a Louvain clustering [22] and modularity (community detection) algorithm, at least two different clusters emerge. Accounts coloured in dark blue in the bottom left of the network are centred around the account of @tagesschau, Germany's major public-service national and international television news service. Similar to the accounts in the centre of the network (in light blue colours), users retweet news and governmental updates on COVID-19 in Germany. However, accounts in the top-right cluster (in orange) have mainly shared tweets in which #CoronaVirusDE had been used for questioning the existence of COVID-19 or the necessity of measures taken against the virus. Many of these accounts are partisans of #AFD, as indicated in red. The strong clustering of the #CoronaVirusDE retweet network (same holds for #FlattenTheCurve) indicates a confirmation of hypothesis (2); retweet networks of COVID-19 related hashtags show a polarisation of users.

Based on the distribution of users in the retweet accounts of #CoronaVirusDE and #FlattenTheCurve we calculate conditional odds of 'hashjacking', i.e., the likeliness that an account in the contra-cluster of a given COVID-19 hashtag is part of a given partisan network, as shown in Fig. 3(A). Here, we see that partisans of #AFD are five times more likely to occur in the contra-cluster of #CoronaVirusDE and 1.5 times more likely to be part of the contra-cluster of #FlattenTheCurve than a user chosen at random. On the contrary, partisans of other party hashtags are less likely to appear in either of the contra-clusters.

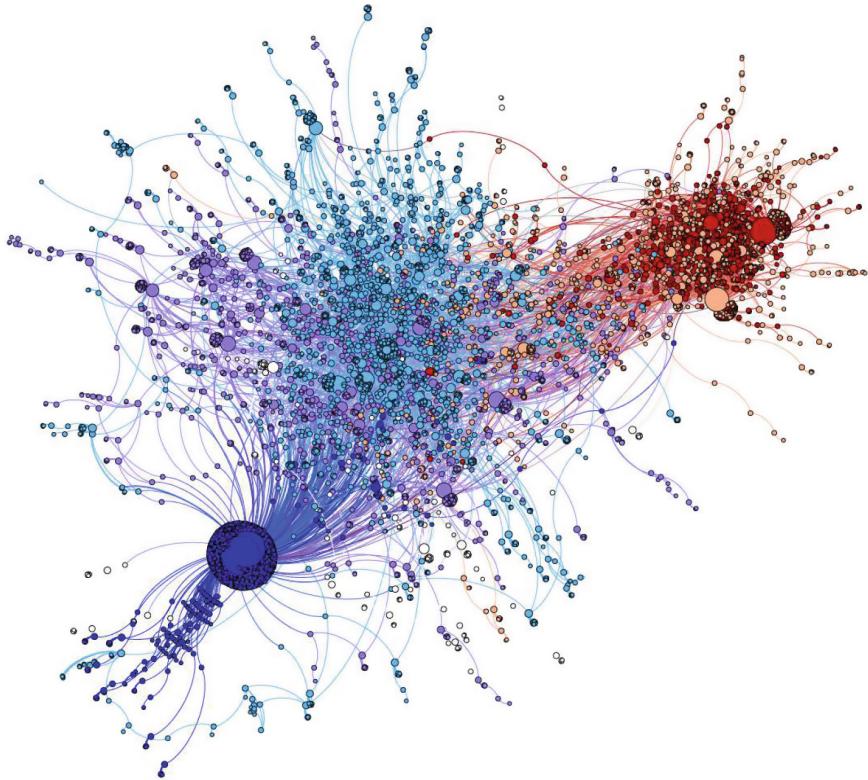


Fig. 2. In April 2020, users of the hashtag `#CoronaVirusDE` are clearly clustered. Users top-right clusters (orange) have shared tweets questioning the existence of COVID-19 or the necessity of measures taken against the virus. 42% of these users and 69 of the 100 most active retweeters are partisans of the hashtag `#AFD` (highlighted in red). We estimate the odds that a user retweeting `#CoronaVirusDE` in critical tweets is a `#AFD` partisan: $Odds(X \in \text{Contra } \#CoronaVirusDE \mid X \in \#AFD \text{ partisan})$.

This finding delivers evidence confirming research hypothesis (3); a significant ‘hashjacking’ activity of right-wing partisans, which is higher than for partisan groups of other political party hashtags.

Lastly, we examine the activity distribution of individual partisan accounts. Figure 3(B) shows accounts across partisans by their activity of retweeting and being retweeted. For all partisan groups, we see a large share of users with very little activity (less than one retweet). However, the distribution of `#AFD` partisans appears to be extremely long-tailed; it contains a set of heavy users. In fact, 39% of all `#AFD` partisan retweets in our sample reference one percent of all `#AFD` partisan accounts. Similarly, 50% of all retweets stem from ten percent of the most active users. The activity (retweeting) of far-right partisans is higher than for partisan groups of other political party hashtags.

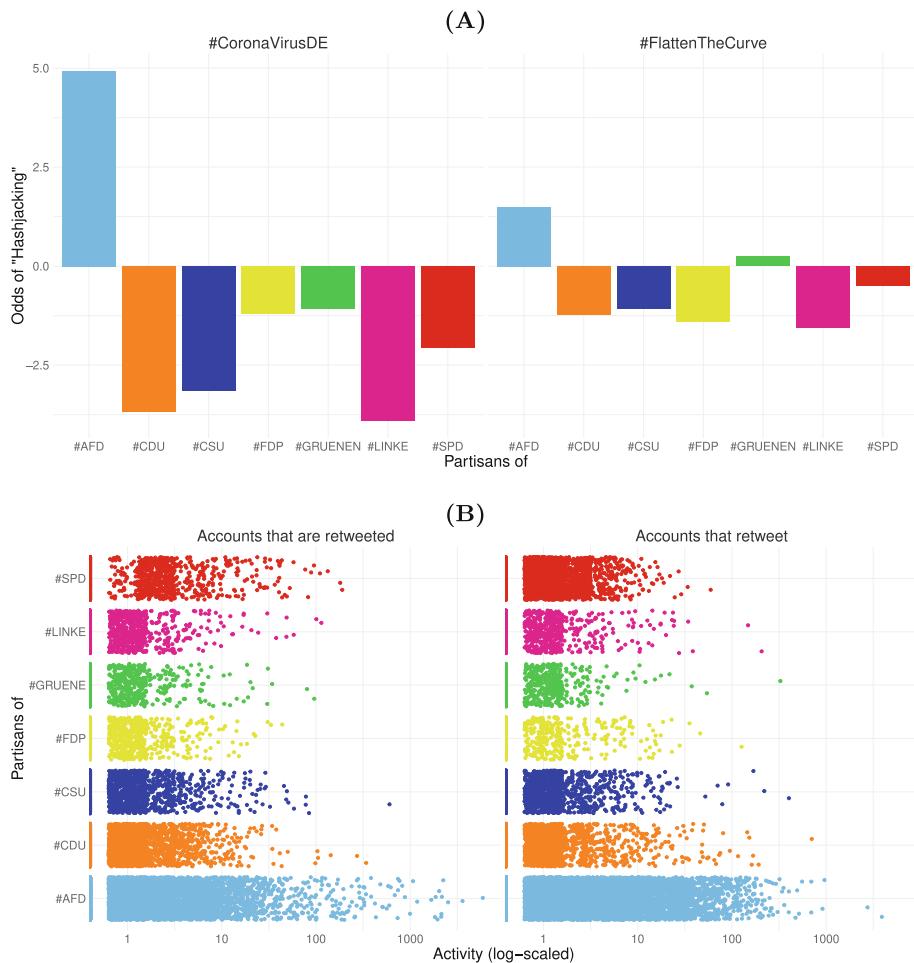


Fig. 3. **(A)** For the retweet networks of #CoronaVirusDE and #FlattenTheCurve, partisans of #AFD are five times, respectively 1.5 times more likely to appear in a contra-cluster than a user chosen at random. Partisans of other major German political party hashtags are less likely to occur in a contra-cluster. **(B)** The group of #AFD partisans contains a set of heavy users. 39% of all #AFD partisan retweets in our sample refer to one percent of #AFD partisan accounts. Similarly, 50% of all retweets stem from the ten percent most active users.

Many of these heavy users contribute to the polarisation of COVID-19 related hashtags, like #CoronaVirusDE. The #AFD partisan accounts in the network in Fig. 2 are highlighted in red. Their dominance in the sceptic contra-cluster (orange) is clearly visible. 42% of all users in the #CoronaVirusDE contra-cluster are #AFD partisans. In fact, 69 of the 100 most active users (in terms of retweets) in this cluster are partisans of #AFD.

6 Conclusion

In this study we investigate the discourse polarisation on Twitter as a result of far-right partisans' application of a hashjacking strategy. For the case of COVID-19 related hashtags and German major political party hashtags, we find that partisans of the "Alternative für Deutschland" (#AFD), systematically polarised discussions on #CoronaVirusDE or #FlattenTheCurve. Similarly, we find that polarisation on political party hashtags in Germany did not change much between 2018 and 2020. But for the exception of the Socialdemocrats' hashtag (#SPD), with a growing opposition cluster, polarisation remains highest for the far-right. #AFD partisans are much more likely to engage in retweeting on COVID-19 related hashtags questioning the existence of COVID-19 or the necessity of measures taken against the virus. At the same time, we highlight that the activity of far-right partisans is much driven by a handful of very active users. 50% of all retweets among #AFD partisans stems from the ten percent most active accounts.

Our work validates the assumption of a high polarisation of debates related to COVID-19. Furthermore, we find evidence that this polarisation is strategically driven by partisans of the far-right. It takes only a small set of very active accounts to polarise retweeting debates on hashtags such as #CoronaVirusDE or #FlattenTheCurve. For users, it is important to be aware of these dynamics of social media discourse, when engaging in it. For platform providers, we show that the critical monitoring of a selected set of accounts can help to contain the spread of misinformation at large.

7 Appendix

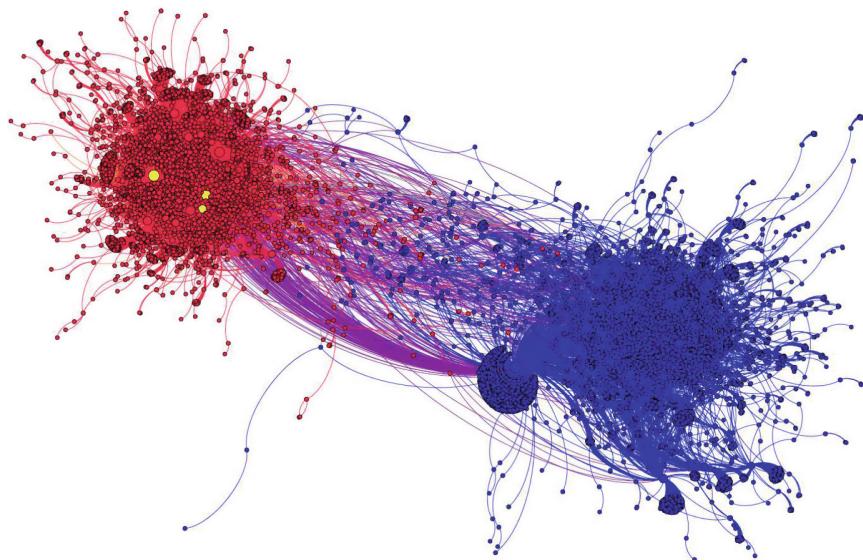


Fig. 4. The retweet network of #AfD from May 2020 shows a clear bimodal clustering of accounts. Accounts in the left (red) cluster are assigned as AfD partisans. The party's official account (@AfD), the AfD parliamentary account (@AfDimBundestag), and the account of the parliamentary leader (@Alice_Weidel) are highlighted in yellow.

Table 1. For thirty randomly selected retweets cluster assignment and content sentiment towards AfD are compared. A Pearson correlation coefficient of 0.92 indicates that cluster assignment and content's sentiment are well aligned.

Tweet id	Assigned cluster	Sentiment towards AfD
1001006481305690000	Pro AfD	+
1001192437107260000	Anti AfD	-
1002961768178730000	Anti AfD	-
1003348201141720000	Anti AfD	-
1002683871828360000	Pro AfD	+
1002902201046650000	Pro AfD	+
1003603237407190000	Pro AfD	+
1001894549344710000	Anti AfD	-
1001431016492410000	Anti AfD	-
1001068784654910000	Pro AfD	+
1001512598989350000	Anti AfD	-
1001185575809010000	Anti AfD	-
1003495411330480000	Pro AfD	?

continued

Table 1. continued

Tweet id	Assigned cluster	Sentiment towards AfD
1002186342430880000	Pro AfD	+
1001940030473030000	Pro AfD	?
1002228287425400000	Pro AfD	+
1003273747158220000	Pro AfD	+
1002468422477800000	Pro AfD	+
1001153121622600000	Pro AfD	+
1001034562187780000	Anti AfD	-
1002919550197870000	Anti AfD	-
1001013938966810000	Pro AfD	+
1003239306176220000	Pro AfD	?
1002163520484540000	Pro AfD	-
1001754476695420000	Anti AfD	?
1002086300806250000	Anti AfD	-
1002370301802440000	Pro AfD	?
1002804299242650000	Anti AfD	-
1000957345516610000	Anti AfD	-

References

1. Gillespie, T.: *Custodians of the Internet*. Yale University Press (2018)
2. Engesser, S., Ernst, N., Esser, F., Büchel, F.: Populism and social media: how politicians spread a fragmented ideology. *Inf. Commun. Soc.* **20**(8), 1109–1126 (2017)
3. Mudde, C.: The populist zeitgeist. *Govern. Opposition* **39**(4), 541–563 (2004)
4. Stier, S., Posch, L., Bleier, A., Strohmaier, M.: When populists become popular: comparing Facebook use by the right-wing movement Pegida and German political parties. *Inf. Commun. Soc.* **20**(9), 1365–1388 (2017)
5. Krämer, B.: Populist online practices: the function of the Internet in right-wing populism. *Inf. Commun. Soc.* **20**(9), 1293–1309 (2017)
6. Frye, C.E.: Carl Schmitt's concept of the political. *J. Politics* **28**(4), 818–830 (1966)
7. Mouffe, C.: Carl Schmitt and the paradox of liberal democracy. *Can. J. Law Jurisprud.* **10**(1), 21–33 (1997)
8. Conover, M.D., Ratkiewicz, J., Francisco, M., Gonçalves, B., Menczer, F., Flammini, A.: Political polarization on Twitter. In: Fifth International AAAI Conference on Weblogs and Social Media (2011)
9. Conover, M.D., Gonçalves, B., Flammini, A., Menczer, F.: Partisan asymmetries in online political activity. *EPJ Data Sci.* **1**(1), 1–19 (2012). <https://doi.org/10.1140/epjds6>
10. Mirer, M.L., Bode, L.: Tweeting in defeat: how candidates concede and claim victory in 140 characters. *New Media Soc.* **17**(3), 453–469 (2015)
11. Hadgu, A.T., Garimella, K., Weber, I.: Political hashtag hijacking in the US. In Proceedings of the 22nd International Conference on World Wide Web, pp. 55–56, May 2013
12. VanDam, C., Tan, P.N.: Detecting hashtag hijacking from Twitter. In Proceedings of the 8th ACM Conference on Web Science, pp. 370–371, May 2016
13. Knüpfer, C., Hoffmann, M., Voskresenskii, V.: Hijacking MeToo: transnational dynamics and networked frame contestation on the far right in the case of the '120 decibels' campaign. *Inf. Commun. Soc.* 1–19 (2020)

14. Darius, P., Stephany, F.: “Hashjacking” the debate: polarisation strategies of Germany’s political far-right on Twitter. In: Weber, I., et al. (eds.) Social Informatics. SocInfo 2019. LNCS, vol. 11864, pp. 298–308. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34971-4_21
15. Bruns, A., Burgess, J.: The use of Twitter hashtags in the formation of ad hoc publics. In: Proceedings of the 6th European Consortium for Political Research (ECPR) General Conference 2011, pp. 1–9. The European Consortium for Political Research (ECPR) (2011)
16. Bruns, A., Moon, B., Paul, A., Münch, F.: Towards a typology of hashtag publics: a large-scale comparative study of user engagement across trending topics. *Commun. Res. Pract.* **2**(1), 20–46 (2016)
17. Pond, P., Lewis, J.: Riots and Twitter: connective politics, social media and framing discourses in the digital public sphere. *Inf. Commun. Soc.* **22**(2), 213–231 (2019)
18. Xu, W.W.: Mapping connective actions in the global alt-right and Antifa counter-publics. *Int. J. Commun.* **14**, 22 (2020)
19. Jacomy, M., Venturini, T., Heymann, S., Bastian, M.: ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. *PloS One* **9**(6), e98679 (2014)
20. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech.: Theory Exp.* **2008**(10), P10008 (2008)
21. Neuendorf, K.A.: The Content Analysis Guidebook. Sage (2017)
22. De Meo, P., Ferrara, E., Fiumara, G., Provetti, A.: Generalized Louvain method for community detection in large networks. In: 2011 11th International Conference on Intelligent Systems Design and Applications, pp. 88–93. IEEE, November 2011



Models of Influence Spreading on Social Networks

Vesa Kuikka^(✉) and Minh An Antti Pham

Finnish Defence Research Agency,
Tykkikenttätie 1, PO BOX 10, 11311 Riihimäki, Finland
vesa.kuikka@mil.fi

Abstract. Complex contagion (CC) and simple contagion (SC) models have been used to describe influence and information spreading on social networks. We investigate four different models, two from each category, and discuss the interplay between the characteristics of social interactions and different variants of the models. Different properties of social networks are discovered by calculating centrality measures with the proposed spreading models. We demonstrate the results with the Zachary's Karate Club social network and a larger Facebook social media network.

Keywords: Influence spreading model · Complex contagion · Simple contagion · Out-centrality · In-centrality

1 Introduction

Complex contagion (CC) and simple contagion (SC) models [3] describe influence and information spreading processes on networks. We present a model for both of these basic cases. We present two variants of complex contagion models. We discuss alternative ways to implement the models: analytical and simulation methods. Flexibility to incorporate additional features to the model and the requirement for computer resources can be different. Analytical and simulation methods of implementing the theory can be guided by these kinds of comparisons. The empirical Zachary's Karate Club social network [18] and a Facebook social media network [13] are used to demonstrate the results of different models.

We model the social networks of individuals as nodes and their interactions as links with weight. We assume that, in the spreading process, nodes in the network change their state from their initial state to an influenced state. Thus far, models of social and biological contagion have been classified into two categories: simple and complex contagion. Simple contagion describes spreading processes induced by a single exposure while complex contagion requires multiple exposures for transmission [3]. Recently, the need for generalised contagion models has been discussed and a unifying model has been presented [15].

In this study, we discuss different network spreading models based on two properties of flow processes, state-dependency and the possibility of loops. We

refer to the concept of a node's state when we consider the probability of spreading through the node. If loops are allowed, one node can be visited several times during the spreading process. If loops are not allowed, we say that only self-avoiding paths are possible. In this study, we categorise the case of state-dependent processes and self-avoiding paths as simple contagion (SC) processes. All other alternatives are considered complex contagion (CC) processes.

Complex networks have been studied with simulation methods and percolation theory, for example. Related to our work are also studies based on random walks on complex networks [16, 17]. An analytical model for studying detailed network structures has been proposed [10, 11]. Although the model can be expressed, in theory, with closed-form expressions, a computer program is necessary to perform the actual calculations.

In the following section, we introduce related work and concepts in the literature. Next, we present our novel influence spreading models and centrality measures computed according to these models. In the context of our analytical spreading models and corresponding centrality measures, we refer to our previous work [6, 10, 12] where pseudo-algorithms of these models have been published. Two simulation models describing simple contagion and diffusion processes are presented, for the first time, in this study. Therefore, we will present these two models and their algorithms in more detail. Zachary's Karate Club social network [18] and a Facebook social media network [13] are used to demonstrate the models. Lastly, we express some general thoughts of the project and present the main conclusions.

The aim of this study is rather to raise questions than solve a specific research gap. This work can be characterised more as an exploratory project than a systematic study of a topic discussed earlier in the literature.

2 Simple and Complex Contagion Models

A survey of models, methods and evaluation aspects related to social influence analysis has been presented in [14]. The authors consider microscopic and macroscopic social influence models. As a future research direction, they suggest that macroscopic models should focus on how to consider human behaviours and different mechanisms during influence spreading processes.

Borgatti [2] has suggested a categorisation of flow processes based on methods of spread and possible trajectories that traffic may follow. In an influence process, individuals affect each other's beliefs or attitudes. The attitudes spread through replication where the trajectories, followed by the attitude, can revisit nodes. One can continue to influence others about the same thing over time [2].

Threshold and cascade models for the diffusion of innovation and maximising the spread of influence through a social network has been studied in [8]. Studies of diffusion processes in the context of mobile telecommunication networks [7] and micro-blog networks [5] are also related to our work. Epidemic spreading and social contagion processes can be described with a variety of activation and diffusion models [1]. Recently, a long-standing problem of how to model belief propagation in systems with short loops has been solved [9].

Complex contagion [4] is the phenomenon in social networks in which multiple sources of reinforcement are required before an individual adopts the change of behaviour. It differs from simple contagion in that it may not be possible for the influence to spread after only one incident of contact with an infected neighbour. The spread of complex contagion across a network of people may depend on many social factors, for instance, how many of one's friends adopt the new idea. The authors in [4] suggest different mechanisms of complex contagion. Emotional contagion is one of the mechanisms that explain the need for multiple exposures in the spread of influence. Centola's work builds on threshold models of collective behaviour [3].

3 Influence Spreading Models

We have studied four different spreading models in the context of influence and information spreading on social networks. Two of them can be described as simple contagion (SC) models where only self-avoiding paths are allowed and nodes transfer influence to their neighbouring nodes only after their state changes. In this study, only the case of self-avoiding paths and state-dependent spreading is regarded as an SC process. All other alternatives are categorised as complex contagion (CC) processes. In our approach, it is typical for node pairs to repeat interactions in the complex contagion models.

The SC model is implemented in the computer program called Sim-sc and the CC model is implemented in the computer program called Alg-cc. Sim-sc and Alg-cc are suggested for modelling information transfer and influence spreading, respectively. These are indicated by the green colour in Table 1.

However, there are different variations of SC and CC models. In Table 1, two examples of such implementations are called Diff-sc and Alg-ccsa. In Diff-sc a diffusion model is implemented that is suitable for describing microscopic diffusion of influence and behaviour spreading. Alg-ccsa is a variant of Alg-cc with self-avoiding paths. The last combination of state-dependent and non-self-avoiding models needs an interpretation in analytical models. We know that the combination is not possible in simulation models since state-dependency leads to the property of self-avoiding.

Table 2 summarises the properties and main application areas of the four information and influence spreading algorithms of Table 1. Sim-sc and Alg-cc are the most important cases because they implement the basic simple contagion and complex contagion spreading models as defined in the 'Introduction'. Sim-sc describes a random walk spreading process where paths are replicated on nodes to all possible neighbouring nodes with the probability of the link between the nodes.

Alg-cc is based on analytical formulas [10] and it describes step-wise spreading with loops, contrary to Sim-sc, where loops are not allowed. There are also two examples of different variations of the two cases of our study. Alg-ccsa is a variation of Alg-cc that allows spreading through different paths but does not allow revisits on one path. This is a special case of complex contagion. Diff-sc is

an implementation of the simple contagion model and it describes the continual diffusion of ideas and innovations.

Notice that applications of the four models behind the algorithms can not be defined exactly. Table 2 just gives some thoughts of different applications. Categorising different influences, behaviour, information and innovation spreading processes is a developing field of research. In practice, social influence is a combination of different processes on social networks [15]. Table 2 provides also information about the efficiency and adaptability of the algorithms and their current program implementations.

Table 1. Four different models and corresponding algorithms describe influence or information spreading in social networks. Alg-cc and Sim-sc, indicated by the green colour, are basic models for describing complex contagion and simple contagion processes, respectively. Alg-cc is based on an analytical mathematical expression. Alg-ccsa is a variant of Alg-cc that is restricted to self-avoiding paths. Diff-sc describes simple contagion as a diffusion process. Sim-sc and Diff-sc are implemented as simulations in computer programs.

state-independent		state-dependent	
self-avoiding	non-self-avoiding	self-avoiding	non-self-avoiding
Alg-ccsa	Alg-cc	Sim-sc	Diff-sc

Table 2. Properties and main application areas of the four models of Table 1. Main applications are only directional because social interactions include many aspects of different influence processes.

Description	Calculation	State-dependent	Main applications	Efficiency	Adaptability	
Alg-cc	Stepwise spreading with loops	Analytical	No	Influence spreading	Fast	Yes/No
Alg-ccsa	Origin aware version of Alg-cc	Analytical	No	Influence spreading with memory	Slow	Yes/No
Sim-sc	Random walk spreading	Simulation	Yes	Information and epidemic spreading	Slow	Yes
Diff-sc	Continual diffusion	Simulation	Yes	Diffusion of ideas and innovations	Slow	Yes

3.1 Simple Contagion (Sim-sc)

Sim-sc was made to calculate the SC variant of Alg-cc. In Sim-sc, the nodes have two states: influenced and not influenced. There is no midpoint, unlike in Alg-cc. The program runs a simulation many times and counts in how many simulations node i had influence from the source node. Each simulation is independent of previous simulations. After R simulations, the influence count of node i is divided by the number of simulations run in total to get the statistical probability of node i being influenced. Then, the source node is changed to the next node until

every node has been a source node. In our implementation, different source node cases are run in different threads to speed up the program.

In a simulation, there is a source node that influences other nodes via links. The simulation starts by running a graph traversal algorithm on the source node. A random number is generated before continuing to the next node to decide if the next node is influenced. The influencing probability depends on the link weight. If the influencing was successful, the traversal continues normally. If it was unsuccessful or the node had already been influenced, the traversal cannot continue to the node and tries to find another path.

To be sure that the simulation worked, we made Sim-cc, the CC variant of Sim-sc, where state dependency was completely removed, even allowing the traversal algorithm to loop between a node pair. Then, we added path length L . This gave the same results as Alg-cc with trivial imprecision due to the characteristics of simulations. Therefore, we can be sure that Sim-sc works as expected.

The time complexity of the program is $O(NR(N + L))$, where N is the number of nodes, L is the number of links and R is the number of repetitions. The time complexity of a traversal algorithm with an adjacency list is $O(N + L)$, which can be seen in the time complexity of the program.

3.2 Complex Contagion (Alg-cc)

We have published a state-independent complex contagion algorithm (Alg-cc in Table 1) based on analytical mathematical formulas of probability theory [10]. The model is state-independent in the sense that the spreading process over a node is independent of the node's state. Generally speaking, this can describe influence and behaviour spreading but usually not epidemic spreading due to the immunity gained after the infected nodes have recovered. Alternative implementations have been presented as pseudo-algorithms in [6, 10, 12]. An advantage of the model is its performance as networks up to 100.000 nodes can be studied with the model [6]. This is because we need not keep track of the history of the paths during the spreading process. Even larger networks can be studied by limiting computations to shorter path lengths or a sub-set of most influential nodes in the network.

A related algorithm Alg-ccsa is a variant of complex contagion algorithm Alg-cc with self-avoiding paths. A drawback of this algorithm is that the performance in the current computer implementation is poor. Consequently, only small networks of up to 100 nodes can be studied with the program. The model can be applied to specific information spreading processes where a node does not pass the same information repeatedly between two nodes while still allowing passing the same information from a different node to the same node. A variant of Alg-ccsa, that does not allow these kinds of re-joining paths, can be computationally slow. We have not yet implemented this, even though it has applications in the modelling of information spreading.

3.3 Diffusion (Diff-sc)

Diff-sc is a variation of the Sim-sc simulation program. It is mostly the same as Sim-sc, but the implementation of the simulation is different. The number of time steps, denoted by $n \in \mathbb{Z}_+$, is also considered in the program. In each time step, active nodes try to activate an inactive node. If it is successful, the inactive node becomes an active node and starts activating other nodes in the next time step. The later a node is activated, the less influence it will have on other nodes since it has fewer time steps; the time steps do not reset to n . After repeating simulations many times, the program counts the statistical probability based on the activated nodes.

Since the nodes activate many times, the program scales all activation probabilities down with the scaling formula Eq. (2) to avoid inflating the results. The scaling formula was made from the idea that the probability of a source node, activating an adjacent node through a single link, stays the same after n time steps. The idea was easier to express mathematically when we negated the statement in Eq. (1).

The probability of a node not being activated during a simulation equals not being activated during a period of one time step:

$$1 - p_t = (1 - p_{t/n})^n, \quad (1)$$

where p_t is the probability of activation during the whole simulation, $p_{t/n}$ is the probability of activation during a single time step and n is the number of time steps. We then solve the scaled probability, $p_{t/n}$:

$$p_{t/n} = 1 - (1 - p_t)^{\frac{1}{n}} \quad (2)$$

Compared to Sim-sc, Diff-sc takes a longer time to run. The time complexity is $O(NRn(N + L))$. Diff-sc yields lower results compared to the other programs. This is because Diff-sc has a limitation: the n time steps. We also tested reducing the limitation to allow every activated node to try to activate other nodes n times. This led to the same results as in Sim-sc. Thus, Diff-sc can be considered as Sim-sc with the limitation of time steps.

4 Applications

We demonstrate the four algorithms of Table 1 for describing interactions on social networks with two empirical network structures: Zachary's Karate Club social network and Facebook social media network. These two networks are different in many respects. The Zachary's Karate Club social network of Fig. 1 is a small network with 34 nodes and 156 directed links between the nodes. The example network of Fig. 3 composed of connections from Facebook has 4039 nodes and 176.468 directed links.

First, we discuss the results of the Karate Club social network [18] of Fig. 1. Panels A and B in Fig. 2 show Out-centrality values of Alg-cc, Alg-ccsa, Sim-sc and Diff-sc (see Table 1 in the section ‘Influence Spreading Models’) for

link weights $W_l = 0.05$ and $W_l = 0.5$ and also, in Alg-cc and Alg-ccsa, $L = 10$. Panels C and D show the corresponding In-centrality values. The definitions of Out-centrality and In-centrality have been presented in [10]. With link weights $W_l = 0.05$ all the models provide similar results except Diff-sc results are somewhat lower. Results show an increasing trend towards more central nodes of Fig. 1. This trend has almost disappeared with high link weights $W_l = 0.5$. This is a result of high spreading power where individual additional connections are not so significant.

On the other hand, differences in Fig. 2 between the four different models are more distinct with $W_l = 0.5$ than $W_l = 0.05$. Differences between Alg-cc and Alg-ccsa are small except in some nodes that have central nodes in their neighbourhood that enable circular interactions in the network structure (see

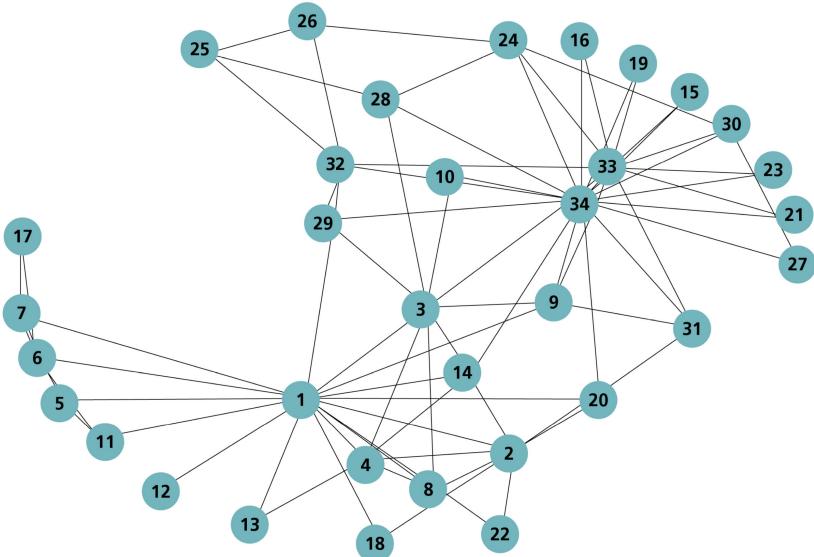


Fig. 1. Zachary's Karate Club social network [18]. Nodes represent members of the club and links relations between them.

Table 3. Computing times of Sim-sc, Diff-sc, Alg-cc and Alg-ccsa for the Zachary's Karate Club social network [18] and the Facebook social media network [13] with two link weights $W_l = 0.05$ and $W_l = 0.5$. In Diff-sc runs, we set time steps to 200. We repeated the simulations 4.000 times. The programs were run with Intel Core i5-8365U processor. Alg-ccsa was not run for the Facebook network because computing times of the current implementation would have been very high.

Zachary's Karate Club social network						Facebook social media network					
$W_l=0.05$			$W_l=0.5$			$W_l=0.05$			$W_l=0.5$		
Sim-sc	Diff-sc	Alg-cc	Alg-ccsa	Sim-sc	Diff-sc	Alg-cc	Alg-ccsa	Sim-sc	Diff-sc	Alg-cc	Alg-ccsa
0.1 s	1.1 s	0.1 s	1.4 s	0.1 s	3.3 s	0.1 s	1.6 s	51 min	7.1 h	52 s	

Fig. 1). Out-centrality and In-centrality values are the same for Sim-sc which is a direct consequence of the definition in section ‘Simple Contagion (Sim-sc)’. Computing times of Sim-sc, Diff-sc, Alg-cc and Alg-ccsa are documented in Table 3 for the Zachary’s Karate Club social network and the Facebook social media network.

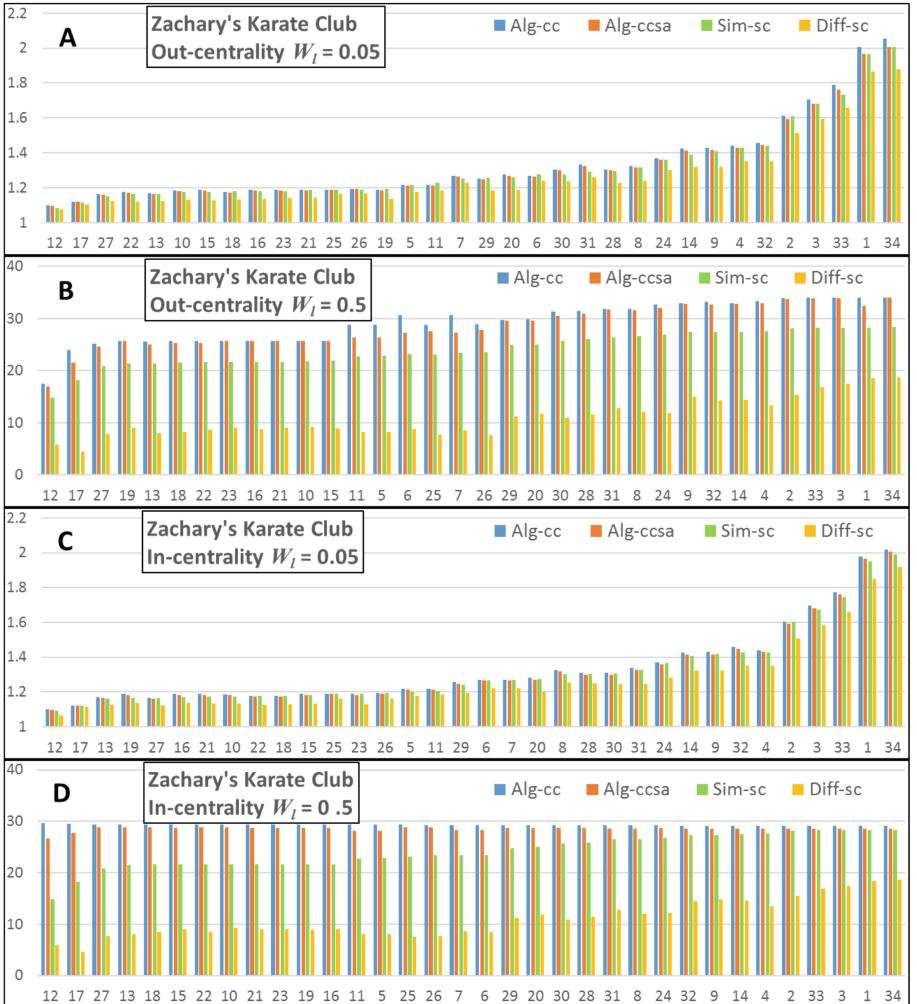


Fig. 2. Values of Out-centrality and In-centrality measures of the Zachary’s Karate Club social network with two link weights W_l . A) Out-centrality $W_l = 0.05$, B) Out-centrality $W_l = 0.5$, C) In-centrality $W_l = 0.05$ and D) In-centrality $W_l = 0.5$. The order of source nodes (horizontal axis) is determined in the ascending order of Sim-sc Out-centrality or In-centrality values. The structure of the network is shown in Fig. 1.

Next, we discuss the results of the Facebook social media network [13]. Panels A and B in Fig. 4 show Out-centrality values of Alg-cc ($L = 8, 11$ and 13), Sim-sc (green) and Diff-sc (yellow) for link weights $W_l = 0.05$ and $W_l = 0.5$. Panels C and D show the corresponding In-centrality values. Notice that nodes of the network on the horizontal axis are ordered in the ascending order of Sim-sc results. As a result of this, Sim-sc results, denoted by the green colour, are displayed as ascending smooth curves in Fig. 4. Again, such like in Zachary's Karate Club network, the Out-centrality and In-centrality values are the same for all nodes in the Facebook network. Alg-ccsa results are not shown because of the low performance of the computer algorithm and program. We discover that the results of the three models are different for the link values of $W_l = 0.05$ as opposed to Zachary's Karate Club in Fig. 2. We conclude that network structures are essential and we should be careful in making general conclusions based on small and simple network structures.

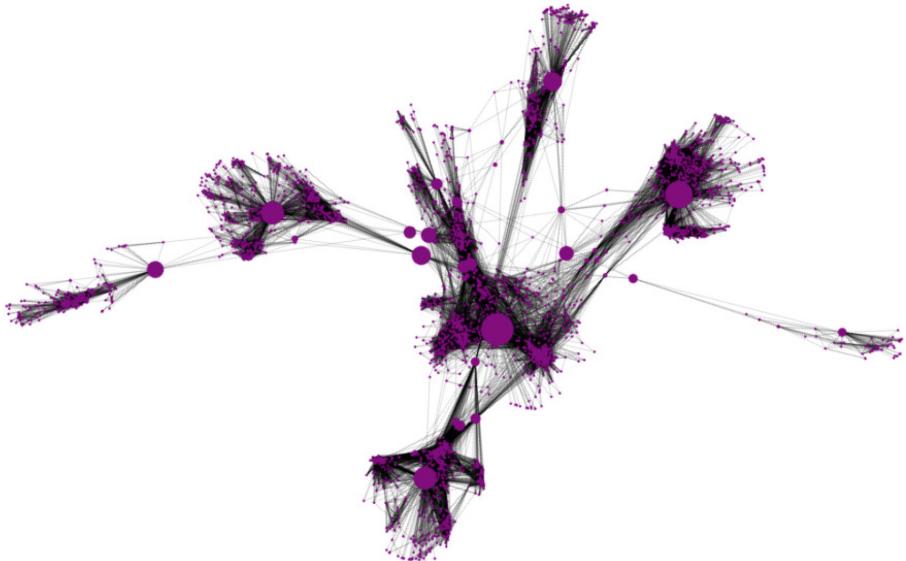


Fig. 3. Facebook social media network [13]. Nodes represent users in the network and links connections between them. Node sizes describe the centrality values of nodes. The graph is generated with the NetworkX Python package.

In addition, the example Facebook network shows some special properties of social media networks. We discover that the structure of this particular Facebook network shows high spreading probabilities in the Alg-cc curves at high maximum path lengths (see Alg-cc Out-centrality results with $L = 8, 11$ and 13 in Fig. 4 A). These effects are due to highly connected clusters (see Fig. 3) that enable loops inside the structure. We conclude this from the fact that Sim-sc results do not show these kinds of influence. These kinds of results in the long

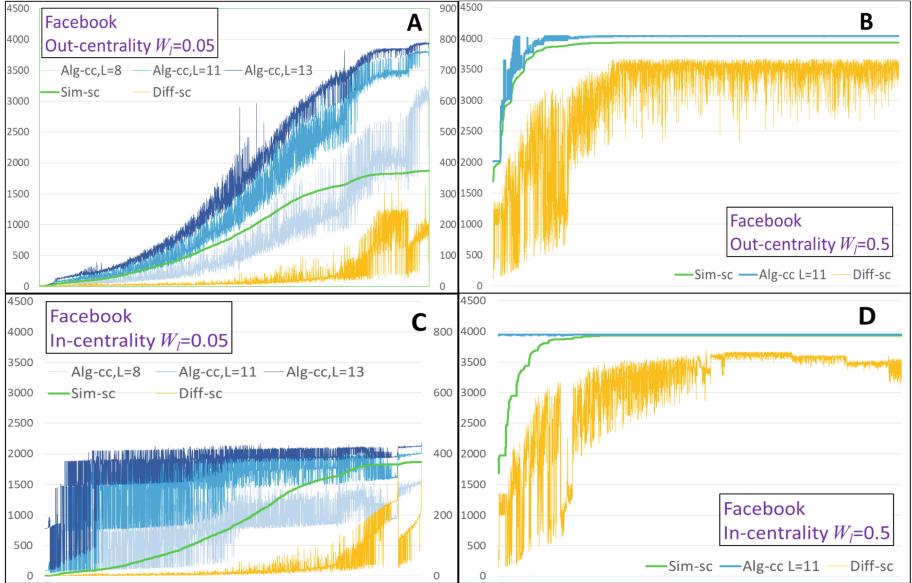


Fig. 4. Values of Out-centrality and In-centrality measures of the Facebook social media network of Fig. 3 with two link weights W_l for Sim-sc, Alg-cc and Diff-sc. A) Out-centrality $W_l = 0.05$, B) Out-centrality $W_l = 0.5$, C) In-centrality $W_l = 0.05$ and D) In-centrality $W_l = 0.5$. Alg-cc results are displayed with the maximum path lengths $L = 8$, $L = 11$ and $L = 13$ for $w_l = 0.05$ and $L = 11$ for $w_l = 0.5$. The order of source nodes (horizontal axis) is determined in the ascending order of Sim-sc Out-centrality or In-centrality values. In A) and C) the scale of Diff-sc is on the right vertical axis.

tails of paths are not possible in small networks like the Zachary's Karate Club social network (see Fig. 2) and usually spreading processes decay more rapidly as a function of the path length. Alg-cc curves in Fig. 4 A with $L = 8$ and $L = 11$ reveal also network structures that are not visible in the Sim-sc curve. Interestingly, these structures can be seen also in the Diff-sc results.

In our earlier work [10] we have investigated the effects of peripheral nodes by comparing our model that allows loops with a model containing only self-avoiding paths. A more detailed study of the results in Fig. 4 could also provide a better understanding of the effects that peripheral nodes have on network structures.

5 Conclusions

In this study, we used two very different empirical network structures: the Zachary's Karate Club social network [18] and a Facebook social media network [13]. We showed that network structures are essential and we should not make too general conclusions about models' predictions and suitability based on small and simple network structures.

In the past, modelling social interactions has been based on two concepts: simple contagion and complex contagion. Recently, a more general idea of competing contagion processes has been proposed [15]. In this study, we have shown that various spreading models exist that describe social interactions. All four models of this study can describe some aspects of influence spreading on social networks.

Our simple contagion model, as defined in this study, has a specific feature that Out-centrality and In-centrality values are the same for all nodes in a network structure if the link weights are the same. This is generally true if link weights are similar and links are bi-directional. These kind of characteristics can be useful in analysing or measuring the properties of spreading processes on networks.

Categorising different types of social interactions is the basis for modelling influence spreading on social networks. The interplay between understanding the phenomena and describing them with mathematical models is necessary. More research is needed in the area of describing different social interactions to further enable the development of mathematical models that are compatible with the descriptions. In practical applications, a combination of models may be needed to describe different aspects of human social behaviour.

Acknowledgements. We are thankful to B.Sc. Juhana Valasmo for his idea of the Diff-sc algorithm in describing diffusion processes on social networks.

References

1. Barrat, A., Barthélémy, M., Vespignani, A.: *Dynamical Processes on Complex Networks*. University Press, Cambridge (2008)
2. Borgatti, S.P.: Centrality and network flow. *Soc. Netw.* **27**(1), 55–71 (2005)
3. Centola, D.: *How Behavior Spreads, The Science of Complex Contagions*. Princeton University Press, Princeton (2018)
4. Centola, D., May, M.: Complex contagions and the weakness of long ties. *Am. J. Sociol.* **113**, 702–734 (2007)
5. Dey K., Lamba H., Nagar S., Gupta S., Kaushik S.: Modeling topical information diffusion over microblog networks. In: Aiello L., Cherifi C., Cherifi H., Lambiotte R., Lió P., Rocha L. (eds.) *Complex Networks and Their Applications VII. COMPLEX NETWORKS 2018. Studies in Computational Intelligence*, **812**, 353–354. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05411-3_29
6. Ijäs, M., Levijoki, J., Kuikka, V.: Scalable algorithm for computing influence spreading probabilities in social networks. In: 5th European Conference on Social Media, Limerick Institute of Technology (ECMS) (2018)
7. Joshi, A., et al.: Social ties and their relevance to churn in mobile telecom networks. In: 11th International Conference on Extending Database Technology (EDBT 2008) (2008)
8. Kempe, D., Kleinberg, J., Tardéos, E.: Maximizing the spread of influence through a social network. *Theory Comput.* **11**, 105–147 (2015)
9. Kirkley, A., Cantwell, G.T., Newman, M.E.J.: Belief propagation for networks with loops. *Sci. Adv.* **7**, 1–15 (2021)

10. Kuikka, V.: Influence spreading model used to analyse social networks. *Comput. Soc. Netw.* **5**, 12 (2018). <https://doi.org/10.1186/s40649-018-0060-z>
11. Kuikka, V.: Modelling community structure and temporal spreading on complex networks. *Comput. Soc. Netw.* **8**(1), 1–38 (2021). <https://doi.org/10.1186/s40649-021-00094-z>
12. Kuikka, V.: Modelling epidemic spreading in structured organisations. *Phys. A* (2021). (Submitted)
13. Leskovec, J., Krevl, A.: SNAP Datasets, Stanford Large Network Dataset Collection (2014). <http://snap.stanford.edu/data>
14. Li, K., Zhang, H., Huang, H.: Social influence analysis: models, methods, and evaluation. *Engineering* **4**, 40–46 (2018)
15. Min, B., Miguel, M.S.: Competing contagion processes: complex contagion triggered by simple contagion. *Sci Rep.* **8**, 10422 (2018)
16. Newman, M.E.J.: A measure of betweenness centrality based on random walks. *Soc. Netw.* **27**(1), 39–54 (2005). <https://doi.org/10.1016/j.socnet.2004.11.009>
17. White, S., Smyth, P.: Algorithms for estimating relative importance in networks. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2003), pp. 266–275 (2003). <https://doi.org/10.1145/956750.956782>
18. Zachary, W.W.: An information flow model for conflict and fission in small groups. *J. Anthropol. Res.* **33**, 452–473 (1973)



Bubble Effect Induced by Recommendation Systems in a Simple Social Media Model

Franco Bagnoli^{1,3(✉)}, Guido de Bonfioli Cavalcabo¹, Benedetto Casu¹, and Andrea Guazzini²

¹ Department of Physics and Astronomy and CSDC, University of Florence,
via G. Sansone 1, 50019 Sesto Fiorentino, Italy

franco.bagnoli@unifi.it

² Department of Education, Languages, Intercultures, Literatures and Psychology,
University of Florence, Sesto Fiorentino, Italy

³ INFN, sez. Firenze, Sesto Fiorentino, Italy

Abstract. We investigate the influence of a recommendation system on the formation of communities in a simulated social media system.

When the recommendation system is initialized with a limited number of random posts, and there is strong selection of correlated users, and therefore communities, not present when examining the internal factors of participating people, arise.

This happens even when people are immutable, but are exposed only to messages coming from people that are correlated to them, according to past messages. When people are let free to evolve, reducing their cognitive dissonance, true isolated communities appear, causing the filter bubble effect.

Keywords: Echo chamber · Filter bubble · Recommendation systems

1 Introduction

Starting from the widespread of Internet and especially when social media like Facebook reached a worldwide spreading, concerns about the effect of recommendation systems arose.

Social media contains such a huge amount of information that simple searches retrieve too many results. Therefore, users are offered selected pieces of information. Among them, there are the posts from people and the pages that the user actively selected to follow, but also those recommended by the system.

There are many methods used by recommendation systems, that we can broadly divide into social engineering-based, content-based and collaborative filtering [13].

Social engineering methods are based on the knowledge of the weakness of our decision systems [9], but also on participation in active discussions, interviews, focus groups etc. Content-based systems are based on the characteristics

of the information disclosed. These systems are quite effective, but they require a certain amount of human intervention.

Collaborative filtering, on the other hand, exploit users' work to extract knowledge from transactions, which in the following are denoted as "posts" (thinking of a discussion on social media). So, from the reading or active evaluation ("likes") of users on posts, one can get information about the similarities among users and recommend unread posts. The same techniques obviously apply to e-commerce.

The drawback of this recommendation system is the possible formation of filter bubbles [12] or echo chambers [5], i.e., the formation of intellectually closed circles of users that exchange information about selected topics. The fear is that separation may lead to the exacerbation of conflicts and radicalism [11].

Although there are many critics to this extreme point of view [4], and many suggestions on how to overcome these barriers [3], a softer consequence of recommendation system is the formation of "artificial" communities that do not reflect the "true" affinity among people, but are just a byproduct of these suggestions.

In order to test this hypothesis, we developed a very simple model of people interacting through a social media, under the influence of a recommendation system.

People are modeled as simple perceptrons [14], i.e., they are characterized by a certain number of factors, and its is the match between these factors and the characteristics of a message that determines the resulting opinion [1]. This approximation is consistent with the usual data analysis in quantitative psychology. Indeed, the idea of representing people as vectors is quite old [16], and it is the (often not reported) basis of factorial analysis [8, 15].

We assume that people emit messages (posts) expressing a subset of their inner factors, since in a single post one does not explore all possible topics. The post is then evaluated by other users who, through the perceptron key-and-lock mechanism, express their opinion on it.

By means of the data series of opinions, it is possible to approximate the similarity among users [2, 10]. However, we are here interested in studying what happens when users are not exposed to all messages or to a random sample of them, but rather only to those recommended by the system based on a previous (limited) sampling.

At first we consider people immutable (they do not change their "tastes"), and in a second part we let them evolve trying to reduce their cognitive dissonance, i.e., "aligning" with the received messages.

2 The Model

Users are represented as vectors of L factors, so that user i corresponds to $U_i = (u_i^{(1)}, u_i^{(2)}, \dots, u_i^{(L)})$, with $-1 \leq u_i^{(k)} \leq 1$. Similarly, posts (messages) are vectors of L components $M_n = (m_n^{(1)}, m_n^{(2)}, \dots, m_n^{(L)})$.

The opinion Ω_{in} of user i about message j is computed as

$$\Omega_{in} = \tanh\left(\frac{\beta}{L} \sum_k u_i^{(k)} m_n^{(k)}\right).$$

The factor β modulates the nonlinearities of the system. If β is large, the system is essentially a Heaviside function, if β is small the system is essentially linear. In the following we shall deal with a linear system:

$$\Omega_{in} = \frac{1}{L} \sum_k u_i^{(k)} m_n^{(k)}.$$

2.1 Recommender Systems

Given the matrix U of users' factors, such that $U_{ik} = u_i^{(k)}$ and a set M of messages $M_{jk} = m_j^{(k)}$, we compute the opinion matrix Ω

$$\Omega = \frac{1}{L} U M^T$$

and from that the user-user Fisher correlation matrix

$$C_{ij} = \frac{1}{Q-1} \sum_{n=1}^Q \left(\frac{\Omega_{in} - E(\Omega_i)}{\sigma(\Omega_i)} \right) \left(\frac{\Omega_{jn} - E(\Omega_j)}{\sigma(\Omega_j)} \right),$$

where Q is the number of messages present in the database, $E(\Omega_i)$ is the average of the Q -vector Ω_i and $\sigma(\Omega_i)$ is the corresponding standard deviation.

Once user i emits a message (post), the recommender system looks for all people that have correlation with i greater than a threshold τ , and exposes them to the message. The corresponding opinions are recorded in the data series. All other people are assigned opinion zero (neutral).

2.2 The Procedure

We generate N users with random factors. We then initialize the database with the opinions of all users on a number Q_0 of random messages.

After that, for a certain (large) number of steps Q_M , one user is drawn at random, and we extract a number of factors (dimensions) by choosing each of them with probability p , so that in average a message covers pL factors. The user therefore emits a message that reflects his/her factors for the chosen number of dimensions. The message is honest and there is no distortion at all.

The message is then proposed to all people that have correlation with the emitter greater than τ , and their opinion is inserted in the database.

If people's factors are let to evolve, we use a parameter ε that models memory: the receiver factors tend to align to the factors appearing in the received message

for a percentage ε , in a way similar to what happens for one user in the Deffuant model [6, 7].

The updated factor k of user i , U'_{ik} due to message m whose components are $m_k = U_{jk}$ with probability p is

$$\begin{cases} U_{ik} + \varepsilon(m_k - U_{ik}) & \text{if } m_k = U_{jk}; \\ U_{ik} & \text{otherwise.} \end{cases}$$

2.3 Cluster Analysis

We used MATLAB function `linkage` to produce dendograms of the overlap among individuals, by varying the cut-off parameter, the function identifies different clusters. This function looks at the pair of items with the shortest euclidean distance, and replaces them with their average, and so on. The dendograms y -axes report the distances.

A reliable community structure corresponds to a separation of clusters for a large interval of the aforementioned parameter, i.e., a large interval in the y -axes where there are no clusters joining. We therefore take the largest “jump” in the y -axes as an indicator of the presence of communities.

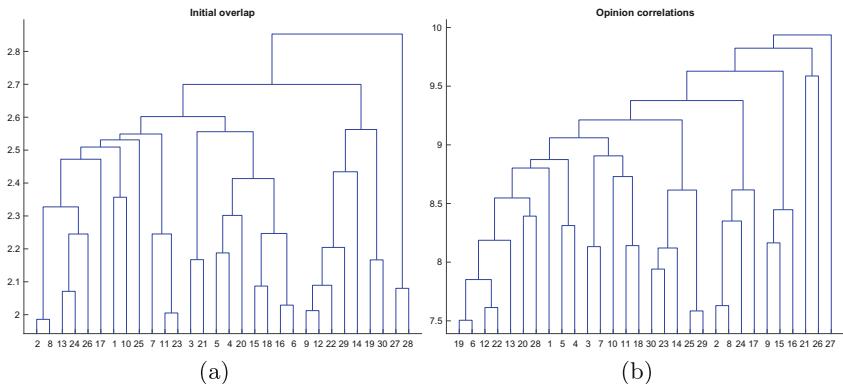


Fig. 1. Community detection with $L = 10$, $Q_0 = 200$, $\tau = 0$, $p = 0.4$, $\varepsilon = 0$, $N = 200$, $Q = 20000$. In the ordinate the cutoff parameter. (a) user overlap, (b) opinion overlap. No evident “jump” is seen, which is reasonable since users are randomly generated and threshold is low.

3 Results

We performed simulations with various values of Q_0 (number of initial messages), p (fraction of factors in emitted messages) and τ (threshold for selecting which user is receiving a recommendation).

When people's factors are not allowed to evolve ($\varepsilon = 0$) no community structures appear in the users overlap (Figs. 1-a and 2-a).

As expected, if we use a large number $Q_0 = 1000$ of initial messages, a low selection threshold $\tau = 0$ and full messages $p = 1$, we get the same one-community as using the real overlap O . This holds also using “shorter” messages ($p = 0.2$) and also reducing the number of initial messages $Q_0 = 200$. In Fig. 1 we report an example.

By increasing the clique threshold $\tau = 0.4$ we see that a community structure appears (Fig. 2).

Fig. 2. Community detection with $L = 10$, $Q_0 = 200$, $\tau = 0.4$, $p = 0.4$, $\varepsilon = 0$, $N = 200$, $Q = 20000$. (a) user overlap, (b) opinion overlap. A higher threshold causes the appearance of a “jump” in the coalescence of cluster, revealing a community structure in the opinion database.

Fig. 3. Community detection with $L = 10$, $Q_0 = 200$, $\tau = 0.4$, $p = 0.4$, $\varepsilon = 0.3$, $N = 200$. (a) initial user overlap, (b) opinion overlap, (c) final user overlap. A higher value of ε completely separates the population.

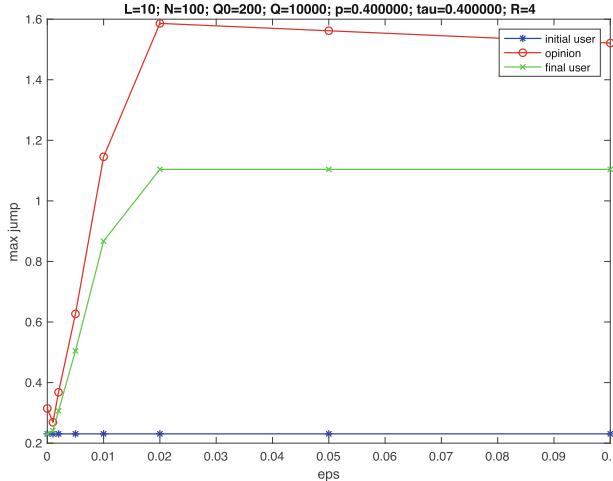


Fig. 4. Community detection with $L = 10$, $Q_0 = 200$, $\tau = 0.4$, $p = 0.4$, $N = 200$, $Q = 10000$. We report the size of the largest jump in the dendrogram as a function of ε . As soon as $\varepsilon > 0$ there is a strong sign of community formation both in the opinions and in the final factors of users. The community structure weakens a bit by increasing ε since in this case the users opinions are more volatile.

If we let user opinion to vary ($\varepsilon > 0$), the communities in the opinion database, that induce the selection of messages, induce the formation of real communities in the final user overlap, Figs. 3 and 4. The faster the adherence to local conformism, the stronger the separation of communities.

4 Conclusions

We have studied the influence of a recommendation system on the formation of originally non-existing communities in a simulated social media system.

In our model, people are modeled as simple linear perceptrons, and are characterized by a set of factors, i.e., preferences or tastes. A message is similarly formed by a set of components, and the opinion about a message is given by the match between factors and components. The message emitted by an individual is formed by a subset of his/her weights. The recommendation system delivers messages only to those people that are expected to express a positive judgment according to the database of past opinions. People then have a certain probability of evolving their weights accordingly.

We start by assigning random weights to people, so there is no initial community structure. When the recommendation system is initialized with a limited number of random posts, apparent communities may appear due to fluctuations in the sampling. When the recommendation system is asked to deliver messages only to people that are expected to express highly positive opinion, those messages are sent only to those belonging to the fake community, which is therefore strengthened in the database.

This effect is even more marked when people are let free to evolve their factors in the “direction” of the incoming messages. In this case real communities form and stabilize due to the interplay between the initial fluctuations and the recommendation system, therefore real isolation bubbles are formed.

Recommendation systems do indeed have an influence on our lives. Probably, the fear for the “inevitable” formation of filter bubbles or echo chambers is exaggerated or not universally applicable, but there is indeed an influence at least in the formation of communities.

The result is that the very first interactions in a social network with a recommendation system may determine an isolation bubble, and even promote the formation of communities or the assignation to a community according to the initial random sampling of messages.

References

1. Bagnoli, F., Berrones, A., Franci, F.: De gustibus disputandum (forecasting opinions by knowledge networks). *Physica A Stat. Mech. Appl.* **332**, 509–518 (2004). <https://doi.org/10.1016/j.physa.2003.09.065>
2. Blattner, M., Zhang, Y.C., Maslov, S.: Exploring an opinion network for taste prediction: an empirical study. *Phys. A Stat. Mech. Appl.* **373**, 753–758 (2007). <https://doi.org/10.1016/j.physa.2006.04.121>
3. Bozdag, E., van den Hoven, J.: Breaking the filter bubble: democracy and design. *Ethics Inf. Technol.* **17**(4), 249–265 (2015). <https://doi.org/10.1007/s10676-015-9380-y>
4. Bruns, A.: Filter bubble. *Internet Pol. Rev.* **8**(4) (2019). <https://doi.org/10.14763/2019.4.1426>
5. Cinelli, M., Morales, G.D.F., Galeazzi, A., Quattrociocchi, W., Starnini, M.: The echo chamber effect on social media. *PNAS* **118**(9), e2023301118 (2021). <https://doi.org/10.1073/pnas.2023301118>
6. Deffuant, G., Neau, D., Amblard, F., Weisbuch, G.: Mixing beliefs among interacting agents. *Adv. Complex Syst.* **03**(01n04), 87–98 (2000). <https://doi.org/10.1142/s0219525900000078>
7. Hegselmann, R., Krause, U., et al.: Opinion dynamics and bounded confidence models, analysis, and simulation. *J. Artif. Soc. Soc. Simul.* **5**(3), 1–33 (2002)
8. Hotelling, H.: Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.* **24**(6), 417–441 (1933). <https://doi.org/10.1037/h0071325>
9. Kahneman, D.: Thinking, Fast and Slow. Farrar, Straus and Giroux, New York (2011)
10. Maslov, S., Zhang, Y.C.: Extracting hidden information from knowledge networks. *Phys. Rev. Lett.* **87**(24), 248701 (2001). <https://doi.org/10.1103/physrevlett.87.248701>
11. O’Hara, K., Stevens, D.: Echo chambers and online radicalism: assessing the internet’s complicity in violent extremism. *Policy Internet* **7**(4), 401–422 (2015). <https://doi.org/10.1002/poi3.88>
12. Pariser, E.: The filter bubble: what the Internet is hiding from you. Penguin Press, New York (2011)

13. Patel, B., Desai, P., Panchal, U.: Methods of recommender system: a review. In: 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS). IEEE (March 2017). <https://doi.org/10.1109/iciiecs.2017.8275856>
14. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **65**(6), 386–408 (1958). <https://doi.org/10.1037/h0042519>
15. Thurstone, L.L.: Multiple factor analysis. *Psychol. Rev.* **38**(5), 406–427 (1931). <https://doi.org/10.1037/h0069792>
16. Thurstone, L.L.: The vectors of mind. *Psychol. Rev.* **41**(1), 1–32 (1934). <https://doi.org/10.1037/h0075959>



Maximum Entropy Networks Applied on Twitter Disinformation Datasets

Bart De Clerck^{1,3(✉)}, Filip Van Utterbeeck¹, Julien Petit¹, Ben Lauwens¹, Wim Mees², and Luis E. C. Rocha^{3,4}

¹ Royal Military Academy, Faculty of Engineering, Department of Mathematics, Brussels, Belgium
declerck.bart@pm.me

² Information, Systems and Sensors, Royal Military Academy, Faculty of Engineering, Department of Communication, Brussels, Belgium

³ Faculty of Economics and Business Administration, Department of Economics, Ghent University, Ghent, Belgium

⁴ Faculty of Sciences, Department of Physics and Astronomy, Ghent University, Ghent, Belgium

Abstract. Identifying and detecting disinformation is a major challenge. Twitter provides datasets of disinformation campaigns through their information operations report. We compare the results of community detection using a classical network representation with a maximum entropy network model. We conclude that the latter method is useful to identify the most significant interactions in the disinformation network over multiple datasets. We also apply the method to a disinformation dataset related to COVID-19, which allows us to assess the repeatability of studies on disinformation datasets.

Keywords: Disinformation · Twitter · Maximum entropy networks

1 Introduction

With the rise of social media, citizens can voice their opinions and interact with each other more than ever. At the same time, there is also an increase in the spread of false or misleading information. According to the definition of the European Commission, disinformation is ‘verifiably false or misleading information created, presented and disseminated for economic gain or to intentionally deceive the public’. Misinformation on the other hand is ‘verifiably false information that is spread without the intention to mislead, and often shared because the user believes it to be true’ [1]. When disinformation is shared by a so-called ‘useful idiot’, it often turns into misinformation, because the person sharing it does not intend to mislead. Democratic elections in several countries have been plagued by both in recent years [10, 21, 24, 25, 32]. During the COVID-19 pandemic, there has been a flare-up of unreliable or low confidence information [16, 17]. Even though it is difficult to quantify the impact and the efficiency of

such campaigns, the social media platforms and different national or supranational entities are actively encouraging research to detect, identify or attribute disinformation.

In this paper we focus on disinformation within the context of information operations: an active operation that is run by an actor with the intent to create desired effects on the will, the understanding and capability of its adversaries or potential adversaries in support of its own objectives. The analysis and identification of coordinated behaviour in social networks has previously been realised by using a user similarity network [29] or by looking at account behaviour and activity patterns [31, 36], sometimes in a limited timeframe [30, 41]. Various tools and platforms exist and contribute to the fight against disinformation, e.g. Hoaxy [35] for the tracking of social news sharing, and various fact checking websites such as FactCheck[2], PolitiFact [3] or EUvsDisinfo [4]. In addition, there are algorithms for identifying artificial accounts [34, 40, 42] and automatically classifying misinformation or propaganda [19].

Currently only a few large social media platforms such as Reddit or Twitter give (sometimes limited) access to their data. On a regular basis, Twitter releases datasets that are suspected to be state-backed information operations (i.e. a disinformation campaign) via the Twitter information operations report [5]. Twitter works in close collaboration with the Stanford Internet Observatory for an independent analysis [11, 13, 14, 18] of the suspected accounts. In these case-by-case studies, both the content and the network topology are analysed. This always includes a subject matter expert who is able to understand the content of the messages and place it within the context of the targeted country.

Unlike other datasets e.g. concerning the 2016 U.S. elections, and despite their availability, the datasets from the Twitter information operations report have not been much studied. One exception is a study that built a semi-supervised ensemble-tree classifier model to detect influential actors in a disinformation network [37]. We use these datasets as a ground truth (curated by the social media platform) to detect disinformation operations.

A common method for analysing the disinformation spread (or information flow) on a social network consists in creating an interaction network (e.g. a retweet or a reply network in the case of Twitter) between different users of the social network and analysing its structure. A community detection algorithm could also be applied to investigate which users are active together. In many instances, the network is binary and non-weighted (due to limitations of algorithms in the tools used), although it has been shown that taking into account the directionality and edge weights can further improve the quality of the detected communities [15, 39]. Even when considering directed and weighted edges, it is possible that the interaction network still contains a lot of noise. Noise refers to the presence of one-time or low-frequency interactions between users that distorts the discovered communities. The noise can be reduced by removing interactions that are not statistically significant using a maximum entropy network [12, 38].

The contributions of this paper are the following: 1) It is the first to apply a maximum entropy network model on the collection of datasets from the Twitter information operations report to identify statistically significant interactions

between users. 2) The technique is validated against existing datasets to retrieve the majority of the interactions in the disinformation network. 3) The analysis of a disinformation dataset regarding COVID-19 provides a critical perspective on repeatability: due to the way in which Twitter data is shared, it can be difficult to replicate studies that use datasets containing misinformation.

2 Methods

2.1 Interaction Networks

We construct a weighted, directed user-user interaction network $G_w = (V, E)$. Each node corresponds to a user and each edge with edge weight w_{ij} represents the amount of times user j interacts (retweets or replies) with user i . The direction of the edge matches the direction of the information flow. We also construct a directed bipartite interaction network $G_{BiP} = (U, P, E)$ [8]. The layer U is composed of Twitter users and the layer P is composed of messages. The edges from U towards P represent authorship whereas edges from P towards U represent an interaction with the message (retweet, reply, quote etc.). The bipartite network can be projected onto the user layer by using \mathcal{V} -motifs. A \mathcal{V} -motif \mathcal{V}_{ij} from user i to user j is defined as the simultaneous existence of an edge from user i to message m and an edge from message m to user j [33]. If \mathcal{V}_{ij} exists, an edge exists from user i to user j in the projection (Fig. 1). In a network we can observe densely connected clusters of nodes. Such a cluster is called a community. Modularity quantifies the quality of a particular division of a network in clusters [28]. Community structure can be detected by optimising the modularity over the possible divisions of a network [27].

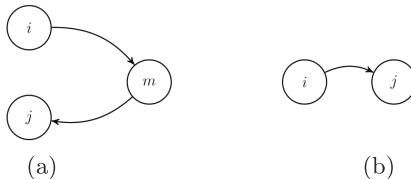


Fig. 1. (a) Bipartite network with a \mathcal{V} -motif from user i to user j . (b) Projection of the \mathcal{V} -motif on the user layer.

2.2 Maximum Entropy Networks

The maximum-entropy method consists of constructing an ensemble of networks whose topology is random, apart from a controlled set of structural constraints. It is used to define null models of a real-world network and patterns are sought for by looking for statistically significant discrepancies between the real network and the null model [38]. For the bipartite network, the Bipartite Directed Configuration Model (BDCM) [20] is a suitable null model. A statistically significant

\mathcal{V} -motif \mathcal{V}_{ij} indicates that user j acts as a significant spreader of user i . The statistically significant motifs are maintained as a user-user interaction on the projection of the network on the user layer. This projection leads to a directed network of information flow between users for each interaction type. We consider each kind of interaction (retweet or reply) separately, because a reply does not necessarily entail support for the user being replied to.

2.3 Toy Example

Figure 2 shows a toy example that illustrates different community detection results for three different network models applied on the same configuration: users 1 through 5 author a large number of messages and have a certain visibility or reputation. Users 6 and 7 aim to artificially increase the visibility of users 1 and 2 by generating many interactions. The remaining users occasionally author a message and also interact among themselves. When using a binary network (i.e. omitting edge directionality and weight), information is lost and the identified communities give a distorted picture (Fig. 2a). Adding edge weight gives better results (Fig. 2b). Filtering out the statistically significant interactions (figure 2c) identifies the ground truth even better. Community detection for this figure was done using the Leiden algorithm [39], as it is capable of working with weighted, directed and even multiplex networks.

In the case where users 1 and 2 author a large proportion of all message and where users 6 and 7 also interact with a large portion of all other messages, their amplification effort would no longer be statistically significant. This can lead to disturbances in the community detections results.

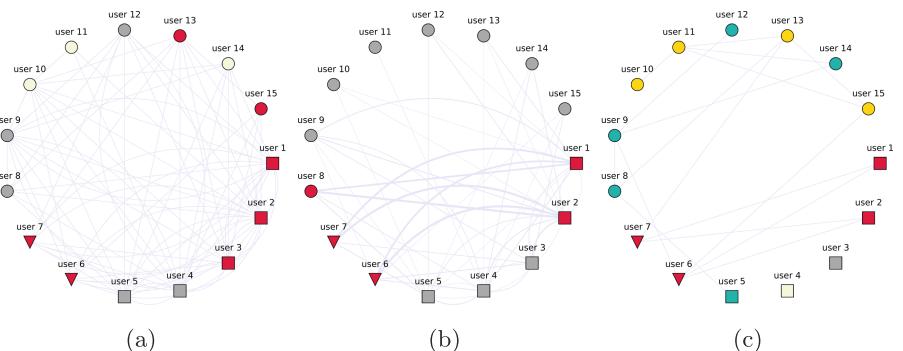


Fig. 2. Community detection in different network representations of the same configuration: users 6 and 7 act as spreaders for users 1 and 2 in a synthetic network. Node shape represents behaviour (square: active poster, triangle: amplifier, circle: normal user). Node colour represents community membership. (a) Binary retweet network. (b) Weighted retweet network. (c) Projected bipartite network.

2.4 Data

The first data source is the Twitter information operations report which contains multiple datasets, linked to a specific disinformation campaign. Each dataset contains information about the users that were identified as connected to the disinformation campaign (flagged users) and their tweets. The tweets with which flagged users interact that are authored by non-flagged users (external tweets) are not included in the dataset, but their tweet id is known. This allows these messages to be downloaded separately to identify which non-flagged users the flagged users interacted with. Table 1 shows the percentage of messages that could be downloaded for each dataset in August 2021.

Table 1. External tweets for each dataset. N_{RT} , $\%_{RT}$ and N_{RP} , $\%_{RP}$ denote the absolute number of messages and the percentages that could be downloaded for retweets and replies respectively.

Country	Period	N_{RT}	$\%_{RT}$	N_{RP}	$\%_{RP}$
Armenia	2021-02	298	68.12	208	41.83
China	2020-05	45,199	27.61	44,048	14.85
Cuba	2020-10	1,225,007	76.98	104,601	62.31
Egypt	2020-04	1,962,519	67.68	353,471	51.41
Honduras	2020-04	222,281	76.28	101,606	57.19
Indonesia	2020-02	367,482	44.66	536,776	36.09
Iran	2020-10	213	66.67	620	50.16
Iran	2021-02	74,070	79.46	79,106	67.37
Qatar	2020-10	128,963	65.38	5,496	48.65
Russia	2020-05	306,980	72.59	449,637	59.53
Russia	2020-10	197	80.20	72	90.28
Russia GRU	2021-02	3,237	64.20	2,592	61.38
Russia IRA	2021-02	20,509	90.17	4,662	56.07
SA, EG & AE	2020-04	7,045,922	53.85	774,943	51.22
Serbia	2020-02	2,589,041	59.01	972,795	40.98
Thailand	2020-10	5,207	91.05	3,814	87.05
Turkey	2020-05	8,513,467	65.15	2,246,872	42.20

The second data source contains Twitter data linked to the release of the documentary ‘Plandemic’, an information operation aimed at spreading disinformation and conspiracy theories. This dataset was constructed from the George Washington University’s publicly available dataset called Tweetsets [22, 26]. In contrast to the tweets from the information operations report datasets, only a fraction (26.48%) of the messages linked to the release of Plandemic could be downloaded. The messages in which the term Plandemic was found are locally

stored in George Washington and are the result of using the STREAM API in conjunction with specific keywords. The Tweetsets webpage only returns tweet ids that need to be downloaded locally. The code used for the data preprocessing, and the analysis in this paper is available on GitHub [6].

3 Results

3.1 Twitter Information Operations Datasets

For both interaction types (retweets and replies), a large part of the edges of the projected user-user network are statistically significant, with some exceptions such as the ‘*russia (2020-10)*’ and the ‘*qatar (2020-10)*’ datasets. Table 2 shows an overview of the results. We consider the method to fail when 1) both the percentage of matched edges in the retweet network ($\%_{M,RT}$) and the percentage of matched edges in the reply network ($\%_{M,RP}$) are less than 60% or 2) one of $\%_{M,RT}$ or $\%_{M,RP}$ is less than 30%. The apparent failure of the method on the dataset ‘*russia (2020-10)*’ is possibly due to the limited size of the network. The large drop in number of retained edges in the ‘*qatar (2020-10)*’ dataset is due to a single flagged user (@ShurafahAlthani) who interacted with more 80% of all

Table 2. Overview of matching edges between the weighted, directed and the projected bipartite network. $\%_{M,RT}$ denotes the number of matched edges for the retweet network. $\%_{M,RP}$ denotes the number of matched edges for the reply network. N denotes the number of nodes in the network.

Country	Period	$\%_{M,RT}$	$\%_{M,RP}$	N
Armenia	2021-02	89.80	58.33	176
China	2020-05	99.89	99.11	59,739
Cuba	2020-10	55.01	80.10	172,073
Egypt	2020-04	93.21	93.96	405,099
Honduras	2020-04	83.90	82.83	51,816
Indonesia	2020-02	65.93	86.47	33,142
Iran	2020-10	75.19	47.77	197
Iran	2021-02	89.48	88.11	15,555
Qatar	2020-10	18.63	27.22	2,764
Russia	2020-05	80.85	93.18	61,938
Russia	2020-10	6.67	6.15	9
Russia GRU	2021-02	69.17	35.16	756
Russia IRA	2021-02	34.18	77.41	1,357
SA, EG & AE	2020-04	84.23	97.41	945,842
Serbia	2020-02	53.58	65.25	1,001,796
Thailand	2020-10	72.32	61.35	2,333
Turkey	2020-05	81.82	90.86	1,069,045

messages. A large portion of flagged users were connected to the disinformation network via this user and are disconnected in the projected network.

We found no correlation between the number of matching edges between the networks and 1) the percentage of external tweets that could be downloaded ($p = 0.17$), 2) the number of external tweets ($p = 0.42$) and 3) the number of nodes in the networks ($p = 0.29$).

Figures 3a and 3b show an illustration for the Honduras retweet network. When running community detection algorithms on both the retweet network and the projected user-user interaction network, we find that the communities and the key players within them are in line with [13]. Although not all edges from the ‘classical’ retweet network are maintained, the majority of flagged users are connected in the projected bipartite network. Because some edges were removed in the projection, the result of the community detection changed slightly, but without changing the overall conclusions on community composition. Similar findings hold for the other datasets with a high percentage of matching edges. With a few exceptions, the removed edges are mainly low weight edges in the weighted network. In the Honduras network for instance, the highest weight of a removed edge is 126 for an edge from @JuanOrlandoH to @tgaparicio. This situation where a \mathcal{V} -motif \mathcal{V}_{ij} is considered non-significant, even with a high weight, can occur when a user j (here @tgaparicio) interacted with a large proportion of all the messages, while at the same time user i (here @JuanOrlandoH) authored a large portion of messages in the dataset.

3.2 Plandemic Dataset

Figure 3c shows the Plandemic retweet network where different communities are intertwined. The result of the method is visible on Fig. 3d where less bridges exist between communities. The same users occupy a central role, but the communities are defined more clearly. Some communities, such as the one around @GiuseppeNoc become isolated from the largest component. In total, 29% of the \mathcal{V} -motifs were not statistically significant. They match the case where a user i authors a large number of messages and user j only has one interaction. A set of interactions with 15 authors account for half of the removed edges in the projected network. The overall modularity increased from 0.90 to 0.97.

4 Discussion

We used a specific maximum entropy network model on two different disinformation data sources to evaluate its usability to extract significant interactions between users in interaction networks. The methodology was successful in the Twitter information operations report datasets that are curated. In the Plandemic dataset, that is less curated, the method was also suitable and led to improved identification of communities in interaction networks.

Social media platforms are making more and more efforts to avoid disinformation on their platforms. This leads to posts or accounts being deleted,

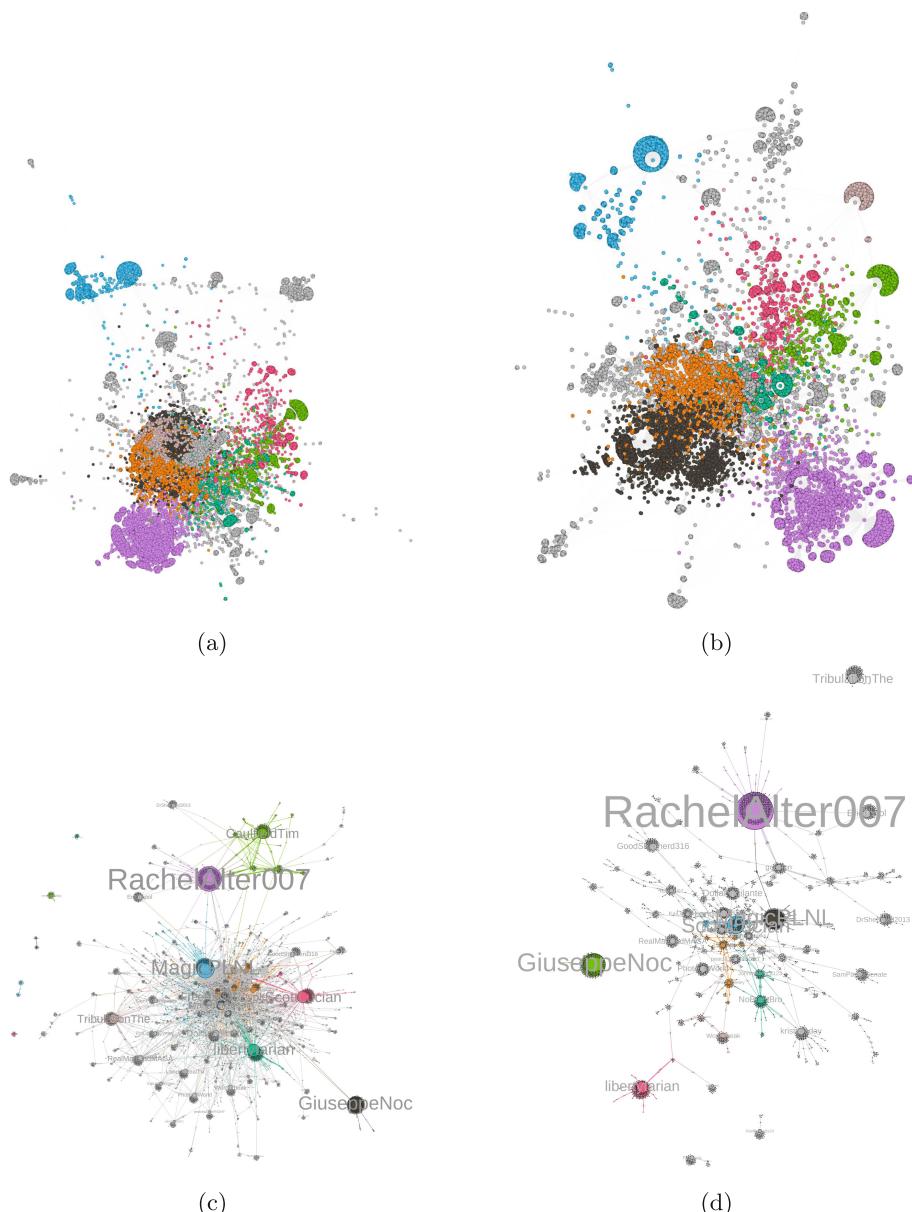


Fig. 3. Community detection results. Node colour indicates community membership. Node size is proportional to node degree. (a) Honduras, classic retweet network. (b) Honduras, projected retweet network (c) Plandemic, classic retweet network. (d) Plandemic, projected retweet network.

often retroactively. Consequently, researchers who wish to conduct a comparative study or to reproduce previous results may run into problems. In the case of Twitter, sharing of collected messages is only allowed on the basis of the ids of the messages that need to be downloaded (rehydrated) afterwards. In the specific case of the Plandemic dataset, among other things, the permanent suspension of the account of the former U.S. president (@realDonaldTrump) following the January 6th insurrection and the removal of a substantial number of messages containing incorrect information about COVID-19 has a major impact on repeatability that increases with time. The accounts of the most central users in the interaction network of the original Plandemic paper [26] no longer exist on Twitter. All datasets we used for this paper suffer from this problem to some extent. At the same time, this also reveals the added value of the datasets from the Twitter information operations report: the deleted accounts and their interactions are provided in their entirety, allowing typical interaction patterns to be studied. Even though some external messages may have been deleted, the researcher still disposes of the data of the most suspicious users in the disinformation campaign.

4.1 Limitations

Because of the limited number of messages that could be downloaded, it was not possible to reconstruct the results of the original Plandemic paper [26]. It would have been interesting to compare the network one obtains with the method from this paper, using a retweet or reply as interaction, with the interaction network from the original paper, where quoting a user in a message was the selected interaction to generate the network.

One of the drawbacks of using the datasets from the Twitter information operations report is that one does not know exactly based on what criteria they were established. Nevertheless, the archive is the only one provided by a large social media platform and it can be used as a ground truth for detecting disinformation operations. In almost every dataset there are a number of accounts that were identified as belonging to the information operation, but who did not author any messages themselves nor interacted with other (flagged) users. These accounts cannot be identified by an interaction-based method. The analysis was performed on a carefully selected part of Twitter activity for which we know a strong suspicion of an organised disinformation operation exists. Thus, a large proportion of the interactions are considered significant. Still, it is relevant to conduct the study because the datasets cover a wide variety of countries and topics. Additionally it would certainly be interesting to analyse to what extent the results remain valid if we were to broaden the scope by collecting all other responses to the messages with which the suspected users have interacted and then re-running the analysis. In practice, however, this is difficult to achieve: the standard Twitter REST API only allows collecting the 100 most recent interactions with a specific tweet. The Twitter Academic Research API [7] overcomes some of these limitations. The best way to obtain these interactions is to collect them as they are created using the STREAM API. This requires the proper

infrastructure and methodology to monitor the social media landscape on a permanent basis.

Although content based analysis has been gaining traction, we limited ourselves to the study of interactions because we try to validate the approach across multiple countries/datasets. Given that the datasets from the information operations report span a multitude of countries and languages, we would require a subject matter expert for each dataset.

For large scale problems, the computation of the \mathcal{V} -motifs and their p -values is costly. Use of a sparse matrix representation in combination with a distributed computing system (or by working on chunks of the data) makes the computation of the expected values of the \mathcal{V} -motifs feasible. The fastLSU algorithm [23], which performs linear scans instead of sorting the p -values can be used as an alternative to the Benjamini-Hochberg procedure [9] used to control the false discovery rate of the significant \mathcal{V} -motifs. It permits separating huge scale testing problems arbitrarily into computationally tractable chunks, similar to what can be done to obtain the \mathcal{V} -motifs.

5 Conclusions and Future Work

The aim of this paper was to analyse the usability of a bipartite null model to extract significant user interactions applied to a series of disinformation datasets. We found this approach to be successful on datasets from the Twitter information operations report. The method can help an analyst reduce the size of the haystack in which to find the needle. It can be included in a pipeline where one integrates other features such as daily user activity, follower count, number of links to external webpages etc.

This paper focused on one specific maximum entropy network model that is usable thanks to simplifications due to the choice of layers. There are other models also suitable for representing interactions on social media. The study of the usability and scalability of such models is the subject of future work. The content of a message also contains relevant information. In a later phase, the approach could be reinforced with content analysis.

In order to increase the speed and quality of research on disinformation dissemination, and given the issues of repeatability, it is worth reflecting on appropriate ways of sharing data consistent with the guidelines of Twitter (or other platforms) to allow new methods to be compared with results from previous studies.

References

1. <https://digital-strategy.ec.europa.eu/en/policies/online-disinformation>
2. <https://www.factcheck.org>
3. <https://www.politifact.com>
4. <https://euvdisinfo.eu>
5. <https://transparency.twitter.com/en/reports/information-operations.html>

6. <https://github.com/B4rtDC/maxentdesinfo>
7. <https://developer.twitter.com/en/products/twitter-api/academic-research>
8. Becatti, C., Caldarelli, G., Lambiotte, R., Saracco, F.: Extracting significant signal of news consumption from social networks: the case of twitter in Italian political elections. *Palgrave Commun.* **5**(1), 91 (2019)
9. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Statist. Soc. Ser. B (Methodol.)* **57**(1), 289–300 (1995)
10. Bradshaw, S., Howard, P.N.: Challenging truth and trust: a global inventory of organized social media manipulation. Technical report, Oxford Internet Institute (2018)
11. Bush, D.: “Fighting like a lion for Serbia”: an analysis of government-linked influence operations in Serbia. Technical report, Stanford Internet Observatory Cyber Policy Center (2020)
12. Caldarelli, G., De Nicola, R., Petrocchi, M., Pratelli, M., Saracco, F.: Flow of online misinformation during the peak of the COVID-19 pandemic in Italy. *EPJ Data Sci.* **10**(1), 34 (2021)
13. Cryst, E., García-Camargo, I.: An analysis of twitter’s takedown of Honduran accounts. Technical report, Stanford Internet Observatory Cyber Policy Center (2020)
14. DiResta, R., Kheradpir, T., Miller, C.: “The world is swimming in a sea of rumors”: influence operations associated with el FAGR newspaper (Egypt). Technical report, Stanford Internet Observatory Cyber Policy Center (2020)
15. Dugué, N., Perez, A.: Directed Louvain: maximizing modularity in directed networks. Ph.D. thesis, Université d’Orléans (2015)
16. Elhadad, M., Li, K.F., Gebali, F.: Detecting misleading information on COVID-19. *IEEE Access* **8**, 165201–165215 (2020)
17. Elhadad, M.K., Li, K.F., Gebali, F.: COVID-19-fakes: A twitter (Arabic/English) dataset for detecting misleading information on COVID-19. In: Barolli, L., Li, K.F., Miwa, H. (eds.) *Adv. Intell. Netw. Collabor. Syst.*, pp. 256–268. Springer International Publishing, Cham (2021)
18. Grossman, S., H., K., DiResta, R., Kheradpir, T., Miller, C.: Blame it on Iran, Qatar, and Turkey: an analysis of a twitter and Facebook operation linked to Egypt, the UAE, and Saudi Arabia. Technical report, Stanford Internet Observatory Cyber Policy Center (2020)
19. Guarino, S., Trino, N., Chessa, A., Riotta, G.: Beyond fact-checking: network analysis tools for monitoring disinformation in social media. In: Cherifi, H., Gaito, S., Mendes, J.F., Moro, E., Rocha, L.M. (eds.) *Complex Networks and Their Applications VIII*, pp. 436–447. Springer International Publishing, Cham (2020)
20. van Lidth de Jeude, J., Di Clemente, R., Caldarelli, G., Saracco, F., Squartini, T.: Reconstructing mesoscale network structures. *Complexity* **2019**, 1–13 (2019)
21. Linville, D.L., Warren, P.L.: Troll factories: the internet research agency and state-sponsored agenda building (2018)
22. Littman, J.: Tweetssets (2018)
23. Madar, V., Batista, S.: FastLSU: a more practical approach for the Benjamini-Hochberg FDR controlling procedure for huge-scale testing problems. *Bioinformatics* **32**(11), 1716–1723 (2016)
24. Marchal, N., Neudert, L.M., Kollanyi, B., Howard, P.N.: Polarization, partisanship and junk news consumption on social media during the 2018 us midterm elections (2018)

25. Mazarr, M.J., et al.: Hostile Social Manipulation?: Present Realities and Emerging Trends. RAND Corporation, Santa Monica, CA (2019)
26. Nazar, S., Pieters, T.: Plandemic revisited: a product of planned disinformation amplifying the COVID-19 “infodemic”. *Front. Public Health.* **9**, 954 (2021)
27. Newman, M.E.J.: Modularity and community structure in networks. *Proc. Natl. Acad. Sci.* **103**(23), 8577–8582 (2006)
28. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. *Phys. Rev. E* **69**(2), 026113 (2004)
29. Nizzoli, L., Tardelli, S., Avvenuti, M., Cresci, S., Tesconi, M.: Coordinated behavior on social media in 2019 UK general election. *CoRR* abs/2008.08370 (2020)
30. Pacheco, D., Flammini, A., Menczer, F.: Unveiling coordinated groups behind white helmets disinformation. *CoRR* abs/2003.01313 (2020)
31. Pacheco, D., Hui, P., Torres-Lugo, C., Truong, B.T., Flammini, A., Menczer, F.: Uncovering coordinated networks on social media. *CoRR* abs/2001.05658 (2020)
32. Woolley, S.C., Howard, P.N.: Computational Propaganda: Political Parties, Politicians, and Political Manipulation on Social Media. Oxford University Press, Oxford (2018)
33. Saracco, F., Di Clemente, R., Gabrielli, A., Squartini, T.: Randomizing bipartite networks: the case of the world trade web. *Sci. Rep.* **5**(1), 10595 (2015)
34. Schuchard, R., Crooks, A., Stefanidis, A., Croitoru, A.: Bots in nets: empirical comparative analysis of bot evidence in social networks. In: Aiello, L.M., et al. (eds.) Complex Networks and Their Applications VII, pp. 424–436. Springer International Publishing, Cham (2019)
35. Shao, C., Ciampaglia, G., Flammini, A., Menczer, F.: Hoaxy: A platform for tracking online misinformation. In: WWW 2016 Companion: Proceedings of the 25th International Conference Companion on World Wide Web (2016)
36. Sharma, K., Ferrara, E., Liu, Y.: Identifying coordinated accounts in disinformation campaigns. *CoRR* abs/2008.11308 (2020)
37. Smith, S.T., Kao, E.K., Mackin, E.D., Shah, D.C., Simek, O., Rubin, D.B.: Automatic detection of influential actors in disinformation networks. *Proc. Natl. Acad. Sci.* **118**(4), 1–10 (2021)
38. Squartini, T.: Maximum-Entropy Networks?: Pattern Detection, Network Reconstruction and Graph Combinatorics. Springer, Cham, Switzerland (2017)
39. Traag, V.A., Waltman, L., van Eck, N.J.: From Louvain to Leiden: guaranteeing well-connected communities. *CoRR* abs/1810.08473 (2018)
40. Wang, M.H., Nguyen, N.L., Dow, C.R.: Detecting potential cyber armies of election campaigns based on behavioral analysis. In: Aiello, L.M., et al. (eds.) Complex Networks and Their Applications VII, pp. 437–446. Springer International Publishing, Cham (2019)
41. Weber, D., Neumann, F.: A general method to find highly coordinating communities in social media through inferred interaction links. *CoRR* abs/2103.03409 (2021)
42. Yang, K.C., Varol, O., Hui, P.M., Menczer, F.: Scalable and generalizable social bot detection through data selection. *Proc. AAAI Conf. Artif. Intell.* **34**(01), 1096–1103 (2020)



Community Deception in Networks: Where We Are and Where We Should Go

Valeria Fionda¹(✉) and Giuseppe Pirrò²

¹ Department of Mathematics and Computer Science,
University of Calabria, Rende, Italy
fionda@mat.unical.it

² Department of Computer Science, Sapienza University of Rome, Rome, Italy
pirro@di.uniroma1.it

Abstract. *Community deception* tackles the following problem: given a target community \mathcal{C} inside a network G and a budget of updates β (e.g., edge removal and additions), what is the best way (i.e., optimization of some function $\phi_G(\mathcal{C})$) to perform such updates in a way that \mathcal{C} can escape to a detector D (i.e., a community detection algorithm)? This paper aims at: (i) presenting an analysis of the state-of-the-art deception techniques; (ii) evaluating state-of-the-art deception techniques; (iii) making available a library of techniques to practitioners and researchers.

Keywords: Community deception · Community hiding

1 Introduction

Complex network analysis is a powerful technique to model and analyze interactions between entities in complex systems (e.g., protein networks, social networks, signaling networks) [17]. One of the prominent tasks that can be performed over these networks is community detection, that is, the task of identifying a (non-overlapping) partition of nodes of the network, providing some insights about their structure [7]. Although community detection has received a lot of the attention, the question concerning *what disclosing the community structure of networks can cause to the users* still remains unsolved. As an example, information about users' community membership can be used by governments to block forms of society self-organization [8]. Another example is the case of Bitcoin trading, where communities are used to identify multiple addresses belonging to the same user [15]. This underlines the need to promote (simple) techniques that can be used by the participants to a community that want to remain below the radar of network analysis techniques like community detection. This gave rise to a new strand of research dubbed as *community hiding* [18] or *community deception* [5], which is concerned with devising algorithms that through principled network updates can hide the community affiliations in order, for instance, to preserve user privacy in the case of social networks. In particular, given a target community \mathcal{C} inside a network G and a budget of updates β (e.g., edge

removal and additions) deception techniques investigate the best way to perform such updates in a way that \mathcal{C} can escape to a detector D (i.e., a community detection algorithm). In order to do so, they try to optimize some function $\phi_G(\mathcal{C})$ to increase the level of hiding of \mathcal{C} inside a community structure (set of communities) $\overline{\mathcal{C}} = \{C_1, C_2, \dots, C_k\}$. What this paper wants to contribute is: **(i)** a systematic overview on the state-of-the-art community deception techniques under a general framework; **(ii)** an evaluation of existing techniques on a variety of real-world networks; **(iii)** a Python library¹ that includes state-of-the-art techniques.

The remainder of the paper is organized as follows. Section 2 introduces the community deception problem. Section 3 reviews where we are. Section 4 reports on an experimental evaluation. Section 5 outlines where we should go.

2 Community Deception in Networks

The goal of community deception is to design algorithms to deceive community detection algorithms. In particular, *given a community \mathcal{C} , the goal is to determine a set β of edge updates so that \mathcal{C} will not be discovered by community detection algorithms*. A network $G = (V, E)$ is an undirected graph that includes a set of $n := |V|$ vertices and $m := |E|$ edges. We denote by $\deg(v) = |N(v)|$ the degree of v , where $N(v)$ is the set of neighbors of v . The set of communities (i.e., a community structure), discovered by some community detection algorithm \mathcal{A}_D is denoted by $\overline{\mathcal{C}} = \{C_1, C_2, \dots, C_k\}$; $C_i \in \overline{\mathcal{C}}$ denotes the i -th community. $E(C_i)$ denotes the set of edges that are incident to some nodes in C_i . We distinguish between intra-community edges of the form $(u, v) : \{u, v\} \subseteq C_i$ and inter-community edges of the form $(u, v) : u \in C_i, v \in C_j$. Given a community C_i , $E(C_i)$ (resp., $\widetilde{E}(C_i)$) denotes the set of intra (resp., inter) edges while $E(C_i, u)$ (resp., $\widetilde{E}(C_i, u)$) denotes the same set for a node $u \in C_i$. The degree of a community is denoted by: $\deg(C_i) = \sum_{v \in C_i} \deg(v)$. E^+ (resp., E^-) denotes a set of edge additions (resp., deletions) on G .

Measuring Deception. As we are concerned with algorithm ways to perform community deception, there is the need for some form of deception measure. Given a network $G = (V, E)$, we denote by $\mathcal{C} \subseteq V$ the community that wants not to be detected. If $\mathcal{C} \in \overline{\mathcal{C}}$ we are in the worst case scenario: the target community is completely discovered. On the other hand, if $\mathcal{C} \notin \overline{\mathcal{C}}$ there can be different ways in which members of \mathcal{C} are hidden within $\overline{\mathcal{C}}$. Generally speaking, some desiderata that one would expect a good hiding of \mathcal{C} in $\overline{\mathcal{C}}$ to have are [6]: **(D1): Reachability Preservation:** \mathcal{C} 's members should be reachable from one another to preserve the information flow; **(D2): Community Spread:** \mathcal{C} 's members should be spread in as many communities as possible in $\overline{\mathcal{C}} = \{C_1, C_2, \dots, C_k\}$; **(D3): Community Hiding:** \mathcal{C} 's members should be distributed in the largest communities of $\overline{\mathcal{C}} = \{C_1, C_2, \dots, C_k\}$. There can be different ways of taking into account these desiderata. Fionda and Pirrò [6] defined the *deception score*, which takes them simultaneously into account.

¹ <https://communitydeception.wordpress.com/>.

Definition 1 (Deception Score). Given a network G , a community \mathcal{C} and a structure $\overline{\mathcal{C}} = \{C_1, C_2, \dots, C_k\}$, the community deception score is: $\mathcal{H}(\mathcal{C}, \overline{\mathcal{C}}) =$

$$\left(1 - \frac{|S(\mathcal{C})| - 1}{|\mathcal{C}| - 1}\right) \times \left(\alpha(1 - \max_{C_i \in \overline{\mathcal{C}}} \{\mathcal{R}(C_i, \mathcal{C})\}) + \gamma(1 - \frac{\sum_{C_i \cap \mathcal{C} \neq \emptyset} \mathcal{P}(C_i, \mathcal{C})}{|C_i \cap \mathcal{C}|})\right)$$

where $|S(\mathcal{C})|$ is the number of connected components in the subgraph induced by \mathcal{C} 's members and α and γ , with $\alpha + \gamma = 1$.

In the definition of deception score, \mathcal{R} (*recall*) and \mathcal{P} (*precision*) are defined as follows: $\mathcal{R}(C_i, \mathcal{C}) = \frac{\#\mathcal{C}'s \text{ members in } C_i \text{ found by } \mathcal{A}_D}{|\mathcal{C}|} \forall C_i \in \overline{\mathcal{C}}$ and $\mathcal{P}(C_i, \mathcal{C}) = \frac{\#\mathcal{C}'s \text{ members in } C_i \text{ found by } \mathcal{A}_D}{|C_i|} \forall C_i \cap \mathcal{C} \neq \emptyset$. \mathcal{H} captures *reachability preservation* by the first multiplicative factor in Definition 1. *Community spread* is captured by the first term in the parenthesis. Finally, *community hiding*, is captured by the second term in the parenthesis. The ideal situation is when each $C_i \in \overline{\mathcal{C}}$ contains a small percentage of \mathcal{C} 's nodes. Overall, we have that $\mathcal{H} \sim 1$ if: (i) \mathcal{C} 's nodes are in a single connected component (to fulfill D1) and (ii) each node in \mathcal{C} belongs to a different (to fulfill D2) and large (to fulfill D3) community; $\mathcal{H} = 0$ if (i) each member of \mathcal{C} belongs to a different component or (ii) $\mathcal{C} \in \overline{\mathcal{C}}$.

Along the same lines, Mittal et al. [10] defined two separate ways of quantifying the hiding of \mathcal{C} within $\overline{\mathcal{C}}$. The *community split* (CommS) quantifies how the members of \mathcal{C} are split into $\overline{\mathcal{C}}$; it ranges from 1 (all members in \mathcal{C} remain in one community in $\overline{\mathcal{C}}$) to $|\overline{\mathcal{C}}|$ (all members of \mathcal{C} get distributed into different communities of $\overline{\mathcal{C}}$). The higher the value of CommS, the wider would be the split of the nodes in \mathcal{C} , thereby increasing the deception of the target community. Authors also defined *community uniformity* (CommU). It is obtained by calculating the entropy of target community's nodes present among the communities in $\overline{\mathcal{C}}$. This score ranges from 0 (when all nodes of \mathcal{C} remain in one community of $\overline{\mathcal{C}}$) to $\log |\overline{\mathcal{C}}|$ (when all nodes of \mathcal{C} get distributed into different communities of $\overline{\mathcal{C}}$). We observe that differently from the deception score, these measures take into account the desiderata listed above separately.

Problem Statement. Figure 1 reports a general deception framework. Given a network G , the *Detector* module (implementing a community detection algorithm) analyzes G to discover communities. The underlying assumption that stresses the need for deception techniques is that the disclosure of (part of) \mathcal{C} leads to privacy leaks and should be avoided. The *Deceptor* module (implementing a community deception algorithm) analyzes the network G and suggests a set of edge rewiring involving nodes in \mathcal{C} that help \mathcal{C} 's members to be hidden as a group. To find the best set of edge updates, the *Deceptor* is based on some function to be optimized such as modularity (minimization) as in the case of DICE [18], node safeness (maximization) as for SAFDEC [6], or permanence (maximization) as for NEURAL [10]. After applying the modifications suggested by the *Deceptor* and obtaining a new network G' , the desiderata is that the *Detector* by analyzing G' is no more able to discover \mathcal{C} ; ideally due to the fact that \mathcal{C} 's members are scattered among different communities. In order to quantify the

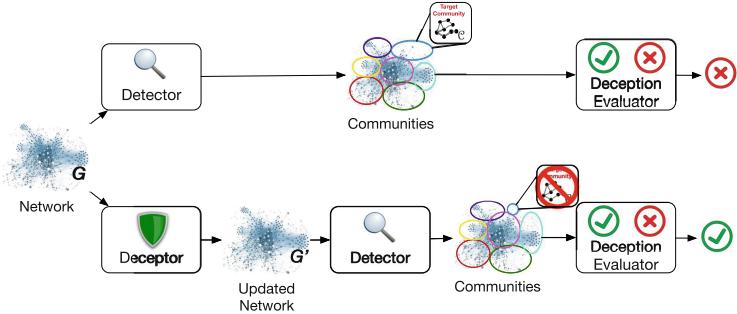


Fig. 1. Community deception: a general framework.

privacy leak caused by the *Detector*, the *Deception Evaluator* module leverages some score such as the *deception score* (Definition 1).

One way to approach the community deception problem would be to work directly with the deception score \mathcal{H} . However, this would require to have knowledge of how the community detection algorithm \mathcal{A}_D , that generated the community structure $\overline{C} = \{C_1, C_2, \dots, C_k\}$ used in the computation of \mathcal{H} , works. What is needed is a way to increase \mathcal{H} by treating a community detection algorithm \mathcal{A}_D as a black-box. To tackle this challenge one can model community deception in terms of the following optimization problem.

Problem 2 (Community Deception). *Given a network $G = (V, E)$, a target community $\mathcal{C} \subseteq V$ and a budget β of updates, solving the community deception problem amounts at solving the following optimization problem:*

$$\underset{G'}{\operatorname{argmax}} \{\phi(G, G', \mathcal{C})\}$$

where $G' = (V, E')$ and $E' = (E \cup E^+) \setminus E^-$ and:

$$\begin{aligned} E^+ &\subseteq \{(u, v) : u \in \mathcal{C} \vee v \in \mathcal{C}, (u, v) \notin E\}, \\ E^- &\subseteq \{(u, v) : \{u, v\} \subseteq \mathcal{C}, (u, v) \in E\}, \text{ and} \\ |E^+| + |E^-| &\leq \beta. \end{aligned}$$

$\phi(G, G', \mathcal{C})$ is a function that models a community deception algorithm while the budget β limits the number of possible updates. The crucial difference between the deception function ϕ and the deception score \mathcal{H} is that the former picks the β changes that maximize ϕ , while \mathcal{H} quantifies (in an axiomatic way) the desirable property that the target community \mathcal{C} is hidden inside $\overline{C} = \{C_1, C_2, \dots, C_k\}$.

3 Community Deception: Where We Are

In this section we survey on existing community deception efforts. We focus on approaches that target the hiding of a target community \mathcal{C} ; other pieces of work like REM [9] that focus on the problem of hiding the whole community structure are orthogonal to our study.

3.1 Modularity-Based Deception

The first community deception approach we describe shows how modularity [12] can be recast for community deception.

Definition 3 (Modularity). *Given a network G , the modularity of the partition of this network into communities $\bar{C} = \{C_1, C_2, \dots, C_k\}$ is given by:*

$$\mathcal{M}_G(\bar{C}) = \frac{\eta}{m} - \frac{\delta}{4m^2} \quad (1)$$

where $\eta = \sum_{C_i \in \bar{C}} |E(C_i)|$ and $\delta = \sum_{C_i \in \bar{C}} \deg(C_i)^2$.

Modularity measures the number of edges falling within groups minus their expected number in an equivalent network with edges placed at random. The objective of many detection algorithms is to maximize modularity [2].

To achieve deception, an intuitive strategy would be to leverage the modularity loss $\mathcal{ML} = \mathcal{M}_G(\bar{C}) - \mathcal{M}_{G'}(\bar{C})$; here, the goal is to find the set of β edge updates where \mathcal{ML} is maximized wrt \mathcal{C} . Maximizing the loss will make the partitioning \bar{C} given by \mathcal{A}_D no more optimal and thus (hopefully) increase the level of hiding of \mathcal{C} . Waniek et al. [18] hinted to a similar approach. Their study suggests a heuristic (called DICE) based on intra-edge deletions and inter-edge additions. However, Waniek et al.'s strategy does not always bring a modularity loss and thus fail to actually contribute to the hiding of the members of \mathcal{C} inside the community structure. Indeed, in some cases the edge update suggested by their strategy may actually increase modularity. To correct for these drawbacks, Fionda and Pirrò [6] proposed a revisited version of DICE (that we refer to as MODMIN) that identifies the best set of updates to minimize modularity wrt the target community \mathcal{C} . In terms of the general deception formulation (see Definition 2) MODMIN instantiates the function ϕ to be the modularity loss $\mathcal{ML} = \mathcal{M}_G(\bar{C}) - \mathcal{M}_{G'}(\bar{C})$. MODMIN adopts a greedy strategy that at each step chooses the edge update that gives the highest \mathcal{ML} . Hence, the goal is to identify updates more profitable in terms of modularity minimization wrt \mathcal{C} [6]:

- **Inter-edge addition:** An inter-community edge addition (u, w) : $u \in C_i \cap \mathcal{C}, w \in C_j$, with $i \neq j$ giving $G' = (V, E \cup \{(u, w)\})$ does not always bring a modularity loss. The highest possible loss is obtained by picking as source and target communities for the edge addition, the communities having the highest degrees. If $\mathcal{C} \in \bar{C}$, the possible modularity loss depends on the rank (in terms of degree) of \mathcal{C} in \bar{C} . If $\mathcal{C} \notin \bar{C}$, the edge insertion with the highest possible loss is (u, w) : $u \in C_i \cap \mathcal{C}, w \in C_j$ and $\deg(C_i) + \deg(C_j)$ is maximal.
- **Intra-edge addition:** An intra-community edge addition (u, w) : $u \in C_i \cap \mathcal{C}, w \in C_i$ giving $G' = (V, E \cup \{(u, w)\})$ brings a modularity loss iff:

$$\frac{\eta - m}{m(m+1)} + \frac{4m^2(\deg(C_i) + 1) - \delta(2m+1)}{4m^2(m+1)^2} > 0 \quad (2)$$

Given an inter-edge addition between communities C_i and C_j (giving G') and an intra-community edge addition in the community C_i (giving the network

G'') we have that: $\mathcal{M}_{G'}(\overline{C}) - \mathcal{M}_{G''}(\overline{C}) = \frac{\deg(C_i) - \deg(C_j) - 2m - 1}{2(m+1)^2}$. Since $\deg(C_i) \leq 2m$, we have that $\mathcal{M}_{G'}(\overline{C}) - \mathcal{M}_{G''}(\overline{C}) < 0$ and, thus, $\mathcal{M}_{G''}(\overline{C}) > \mathcal{M}_{G'}(\overline{C})$ and thus the largest modularity loss can be obtained via an inter-community edge addition.

- **Intra-edge deletion:** For any intra-edge deletion (u, w) : $u \in C_i \cap \mathcal{C}, w \in C_i$ giving $G' = (V, E \setminus \{(u, w)\})$, $\mathcal{ML} > 0$ if and only if:

$$\frac{m - \eta}{m(m-1)} + \frac{\delta(2m-1) - 4m^2(\deg(C_i) - 1)}{4m^2(m-1)^2} > 0$$

Hence, the best edge deletion, in terms of modularity loss, is an intra-community edge in the community C_i having the lowest degree and such that $C_i \cap \mathcal{C} \neq \emptyset$. The modularity loss is the same no matter the pair of nodes $u \in C_i \cap \mathcal{C}$ and $w \in C_i$.

- **Iter-edge deletion:** For any inter-community edge deletion (u, w) : $u \in C_i \cap \mathcal{C}, w \in C_j$, with $i \neq j$ giving $G' = (V, E \setminus \{(u, w)\})$: $\mathcal{ML} > 0$ iff:

$$\frac{\delta(2m-1) - 2m^2(\deg(C_i) + \deg(C_j) + 1)}{4m^2(m-1)^2} - \frac{\eta}{m(m-1)} > 0$$

If $\mathcal{C} \in \overline{C}$, the (possible) modularity loss depends on the rank (in terms of degree) of \mathcal{C} in \overline{C} . If $\mathcal{C} \notin \overline{C}$, the edge deletion with the possible highest loss is (u, w) : $u \in C_i \cap \mathcal{C}, w \in C_j$, with $i \neq j$, where the sum of $\deg(C_i)$ and $\deg(C_j)$ is minimal.

3.2 Safeness-Based Deception

Safeness-based deception [6] has been introduced to correct for some drawbacks of modularity-based deception. In particular, with modularity-based deception one needs to know the entire community structure in order to pick the best edge update (that depends on the degree of the community toward which a new edge should be inserted). Safeness-based deception only requires information that can be obtained from \mathcal{C} 's members.

Definition 4 (Node Safeness). Let $G = (V, E)$ be a network, $\mathcal{C} \subseteq V$ a community, and $u \in \mathcal{C}$ a member of \mathcal{C} . The safeness of u in G is defined as:

$$\sigma(u, \mathcal{C}) := \tau \frac{|V_{\mathcal{C}}^u| - |E(u, \mathcal{C})|}{|\mathcal{C}| - 1} + \chi \frac{|\tilde{E}(u, \mathcal{C})|}{\deg(u)} \quad (3)$$

where $V_{\mathcal{C}}^u \subseteq \mathcal{C}$ is the set of nodes reachable from u passing only via nodes in \mathcal{C} , $E(u, \mathcal{C})$ (resp., $\tilde{E}(u, \mathcal{C})$) is the set of intra- \mathcal{C} (resp., inter- \mathcal{C}) edges, $\eta, \chi > 0$, and $\eta + \chi = 1$.

The leftmost part of Eq. (3) takes into account the portion of nodes in \mathcal{C} that can be reached only via other nodes in \mathcal{C} balanced by the number of intra-community edges. In the ideal situation a member of \mathcal{C} will be able to reach

all the other members of \mathcal{C} with the minimum number of edges, that is, one. This component gives an account of how-well u can transmit information in \mathcal{C} . The second term of eq. (3) gives an account on how u is “hidden” inside the network with respect to its degree. To increase its safeness u should diversify its connections, that is, have the right proportion of links with members of communities other than \mathcal{C} .

Definition 5 (Community Safeness). *Given a network $G = (V, E)$ and a community $\mathcal{C} \subseteq V$, the safeness of \mathcal{C} is defined as: $\sigma(\mathcal{C}) = \sum_{u \in \mathcal{C}} \sigma(u, \mathcal{C}) / |\mathcal{C}|$*

Defining the safeness of \mathcal{C} starting from the safeness of its members allows to identify the least safe members and rewire their links to increase the safeness score of the whole \mathcal{C} . Safeness allows to control different aspects of a community such as reachability and internal/external edge balance. In terms of the general deception formulation (see Definition 2) this approach instantiates the function ϕ to be the safeness gain $\xi_{\mathcal{C}} = \sigma(\mathcal{C}') - \sigma(\mathcal{C})$. It adopts a greedy strategy that at each step chooses the edge update that gives the highest $\xi_{\mathcal{C}}$. Therefore, the goal is to understand what kind of update is more profitable safeness-wise [6]:

- **Intra-edge addition:** An intra- \mathcal{C} edge addition (u, w) s.t. $\{u, w\} \subset \mathcal{C}$ giving an updated network $G' = (V, E \cup \{(u, w)\})$ does not always introduce a safeness gain. The possibility for such an edge to increase the safeness of the community occurs when the edge connects previously disconnected portions of \mathcal{C} . If no new communication path among \mathcal{C} 's members is made available, the new edge will certainly decrease the safeness score.
- **Inter-edge addition:** For any inter- \mathcal{C} edge addition (u, w) s.t. $u \in \mathcal{C}$ and $w \notin \mathcal{C}$ giving $G' = (V, E \cup \{(u, w)\})$ we have that: (i) $\xi_{\mathcal{C}} \geq 0$ always holds; (ii) the maximum increase in safeness happens for the nodes $u \in \operatorname{argmin}\{\frac{|\tilde{E}(u, \mathcal{C})|}{\deg(u)}\}$.
- **Inter-edge deletion:** An inter- \mathcal{C} edge deletion (u, w) : $u \in \mathcal{C}, w \notin \mathcal{C}$ giving $G' = (V, E \setminus \{(u, w)\})$ always gives $\xi_{\mathcal{C}} \leq 0$.
- **Intra-edge deletion:** An intra- \mathcal{C} edge deletion (u, w) : $\{u, w\} \subseteq \mathcal{C}$ does not always bring a safeness gain. In particular, the edge deletion bringing the highest increase in the safeness score is the one for which $\frac{|\tilde{E}(u, \mathcal{C})|}{2\deg(u)(\deg(u)-1)} + \frac{|\tilde{E}(w, \mathcal{C})|}{2\deg(w)(\deg(w)-1)}$ has the maximum value.

3.3 Permanence-Based Deception

We now report on how *permanence*, a node-centric metric used for community deception. Permanence [3] is a vertex-centric metric that quantifies the containment of a node v in a network community C :

$$\text{Perm}(v, G) = \frac{I(v)}{E_{\max}(v)} \times \frac{1}{\deg(v)} - (1 - C_{in}(v)) \quad (4)$$

where $I(v)$, denoted by the internal connections of a node v within its own community, $E_{\max}(v)$ the maximum connections of v to its neighboring communities,

$v, C_{in}(v)$ the fraction of actual and possible number of edges among the internal neighbors of v . This metric indicates that a vertex would remain in its own community as long as its internal pull is greater than the external pull or its internal neighbors are densely connected to each other, hence forming a near clique. The permanence for a network G is then defined as $\text{Perm}(G) = \frac{\sum_{v \in V} \text{Perm}(v)}{|V|}$.

Mittal et al. [10] devised NEURAL, a permanence-based deception strategy, which aims at reducing permanence of the network wrt \mathcal{C} . In terms of the general deception formulation (see Definition 2) NEURAL instantiates the function ϕ to be the permanence loss $\mathcal{P}_l = \text{Perm}(G) - \text{Perm}(G')$ and adopts a greedy strategy that at each step chooses the edge update giving the highest \mathcal{P}_l [10]:

- **Inter-edge addition:** Adding an inter-community edge $\langle u, v \rangle$ where $u \in C$ and $v \in C'$, such that $C \cap C' = \emptyset$, always results in permanence loss. The loss is more if C' is the community that provides the maximum external pull for node u . Therefore, there is Permanence loss in the case of adding an inter-community edge such that $E_{max}(u)$ changes after edge addition.
- **Intra-edge addition:** Adding an intra-community edge $\langle u, v \rangle$ where $u, v \in C$ does not always ensure a loss in Permanence.
- **Intra-edge deletion:** Deleting an intra-community edge $\langle u, v \rangle$ where $u, v \in C$, always results in Permanence loss. The intra-community edge deletion would also affect the Permanence measure for nodes that have both u and v as their neighbors.
- **Inter-edge deletion:** Deleting an inter-community edge $\langle u, v \rangle$ where $u \in C$ and $v \in C'$ such that $C \cap C' = \emptyset$, does not result in Permanence loss.

4 Evaluating Community Deception Algorithms

The goals of the evaluation are to: quantify how deception algorithms can hide a community \mathcal{C} from detection algorithms in real-world networks. Code and datasets are available online².

Experimental Setting. We describe the experimental setting referring to Fig. 1.

Detectors. As detectors, we considered five detection algorithms the code of which is available in the Python version of the `igraph` library: Louvain [1] (`louv`); WalkTrap [13] (`walk`); Greedy [4] (`gre`); InfoMap [16] (`inf`); Label propagation [14] (`lab`).

Deceptors. We considered in the evaluation the following deceptors: **DICE**: the approach by Waniek et al. [18], which randomly adds external edges and deletes internal edges; **MODMIN**: an improved version of DICE discussed in Sect. 3; **SAFDEC**: the safeness-based deception approach; **NEURAL**: the permanence-based deception approach; **NAG**: this approach updates the network by adding edges between nodes selected on the basis of vertex-centrality measures [11]; **RAND**: an approach, which randomly selects both the type of update and the endpoints of the edge addition/deletion.

² <https://communitydeception.wordpress.com/>.

Deceptor evaluator. We ran the experiments in the worst-case scenario, that is, assuming that the target community \mathcal{C} is fully disclosed (i.e., $\mathcal{C} \in \overline{\mathcal{C}}$). The target community \mathcal{C} is picked from the middle of the community distribution for a specific detection algorithm. Then, we apply a budget of changes β such that $|E^+| + |E^-| \leq \beta$ to the initial network and obtain an updated network G' . Then, we run (again) \mathcal{A}_D on G' and compute the values of the deception score on this updated network to measure the impact of the deception strategy.

Experiments have been conducted on a PC i5 CPU 3.0 GHz (4 cores) and 16 GBs RAM. Results reported are the average (95% confidence interval) of 5 runs. We set values of α, γ (for the deception score) to 0.5.

Datasets We considered the following eight real-world networks: **pubmed** (**pub**): it consists of 19,717 scientific publications from PubMed; **webkb**: it contains 877 web pages from four computer science departments; **cora**: it contains 2,708 scientific publications in machine learning, connected by 5,429 citation links; **citeseer** (**cit**): it contains 3,319 publications with 4,722 reference links; (**blog**): it contains 333,983 edges and 10,312 nodes; **facebook** (**fb**): it consists of friends lists from Facebook collected from survey participants; **terrorist** (**terr**): it contains information about terrorists and their relationships; **terrorist-attacks** (**terratt**): This dataset consists of 1293 terrorist attacks each assigned one of 6 labels indicating the type of the attack.

4.1 Deception Score Evaluation

We now report (Fig. 2) on the performance of the deception strategies in terms of deception score. We observe that a random strategy is not effective in hiding \mathcal{C} from none of the detector considered; this comes as no surprise and underlines the need for principled and deception-oriented edge update strategies. Taking into account node popularity (**NAG**) shows some improvement; however, this approach still is not able to obtain the same benefits as from adding/deleting edges from/to \mathcal{C} 's members. We observe that the **DICE** approach (Delete Internal Connect External) performs relatively well on **pub** when considering a budget of updates equals to the 10% of \mathcal{C} 's size while it does not perform equally good in any other configuration. The interesting point is that **MODMIN**, an improved version of **DICE**, consistently looks more promising than **DICE** itself. However, we observe that both approaches, as described in Sect. 3 requires complete knowledge of the community structure in order to pick the best edge update. This constraint is relaxed by **NEURAL**, which performs better than these approaches in all networks considered. However, we notice that from a practical point of view running this algorithm is more time consuming than the other ones. Finally, we observe that **SAFDEC** seems to provide the best performance when it comes to hide \mathcal{C} with different budget values. In general, we observe that the higher the budget the better the deception score.

		Pubmed (pub)													
Deception Score		$\beta = 10\% C $				$\beta = 20\% C $				$\beta = 30\% C $					
		NAG	DICE	MODMIN	NEURAL	SAFDEC	NAG	DICE	MODMIN	NEURAL	SAFDEC	NAG	DICE	MODMIN	NEURAL
louv	.329	.534	.461	.521	.537	.377	.413	.468	.525	.560	.411	.393	.456	.504	.570
walk	.228	.411	.301	.311	.348	.257	.317	.324	.331	.362	.286	.182	.330	.354	.378
gred	.236	.286	.265	.282	.294	.272	.252	.286	.304	.327	.264	.292	.307	.318	.351
inf	.243	.345	.281	.286	.338	.258	.189	.286	.304	.405	.251	.263	.279	.323	.468
lab	.263	.412	.328	.300	.349	.270	.273	.307	.316	.367	.281	.256	.324	.331	.437
RAND						.162				.153					.134
		CORA (cora)													
Deception Score		$\beta = 10\% C $				$\beta = 20\% C $				$\beta = 30\% C $					
		NAG	DICE	MODMIN	NEURAL	SAFDEC	NAG	DICE	MODMIN	NEURAL	SAFDEC	NAG	DICE	MODMIN	NEURAL
louv	.084	.099	.171	.149	.211	.106	.286	.309	.324	.391	.390	.380	.402	.395	.391
walk	.149	.199	.229	.235	.277	.193	.300	.316	.308	.331	.193	.300	.316	.308	.331
gred	.236	.281	.295	.309	.331	.264	.308	.331	.324	.426	.264	.308	.331	.324	.426
inf	.178	.154	.171	.192	.270	.222	.098	.207	.243	.312	.222	.098	.207	.243	.312
lab	.251	.249	.264	.271	.318	.308	.281	.324	.367	.498	.308	.281	.324	.367	.498
RAND						.194				.132					.204
		Blog Catalog (blog)													
Deception Score		$\beta = 10\% C $				$\beta = 20\% C $				$\beta = 30\% C $					
		NAG	DICE	MODMIN	NEURAL	SAFDEC	NAG	DICE	MODMIN	NEURAL	SAFDEC	NAG	DICE	MODMIN	NEURAL
louv	.235	.292	.325	.309	.410	.333	.260	.354	.365	.451	.258	.211	.222	.360	.451
walk	.083	.234	.241	.288	.377	.141	.093	.297	.360	.448	.141	.093	.297	.360	.448
gred	.141	.217	.249	.244	.471	.207	.214	.340	.293	.484	.207	.214	.340	.293	.484
inf	.142	.125	.214	.222	.278	.287	.280	.295	.281	.349	.287	.280	.295	.281	.349
lab	.219	.172	.190	.149	.290	.150	.128	.193	.215	.338	.150	.128	.193	.215	.338
RAND						.115				.292					.236
		WebKB (webkb)													
Deception Score		$\beta = 10\% C $				$\beta = 20\% C $				$\beta = 30\% C $					
		NAG	DICE	MODMIN	NEURAL	SAFDEC	NAG	DICE	MODMIN	NEURAL	SAFDEC	NAG	DICE	MODMIN	NEURAL
louv	.082	.082	.105	.119	.182	.128	.098	.152	.224	.406	.153	.105	.170	.199	.574
walk	.126	.160	.163	.139	.341	.200	.163	.217	.232	.484	.216	.209	.276	.287	.547
gred	.166	.216	.235	.243	.301	.226	.206	.295	.352	.624	.350	.306	.396	.503	.772
inf	.082	.141	.158	.179	.250	.128	.265	.278	.287	.420	.200	.261	.357	.357	.515
lab	.214	.209	.227	.244	.424	.275	.203	.348	.389	.561	.350	.285	.498	.493	.587
RAND						.082				.192					.153
		Citeseer (cit)													
Deception Score		$\beta = 10\% C $				$\beta = 20\% C $				$\beta = 30\% C $					
		NAG	DICE	MODMIN	NEURAL	SAFDEC	NAG	DICE	MODMIN	NEURAL	SAFDEC	NAG	DICE	MODMIN	NEURAL
louv	.177	.165	.191	.244	.314	.373	.303	.353	.373	.473	.349	.287	.490	.507	.556
walk	.161	.162	.201	.329	.358	.346	.098	.353	.456	.443	.404	.356	.395	.465	.556
gred	.094	.220	.244	.235	.375	.223	.267	.355	.292	.476	.404	.351	.428	.476	.545
inf	.091	.081	.125	.167	.194	.213	.303	.408	.356	.464	.319	.341	.421	.518	.553
lab	.177	.227	.251	.245	.291	.220	.214	.303	.313	.376	.350	.306	.262	.357	.478
RAND						.124				.194					.212
		Facebook (fb)													
Deception Score		$\beta = 10\% C $				$\beta = 20\% C $				$\beta = 30\% C $					
		NAG	DICE	MODMIN	NEURAL	SAFDEC	NAG	DICE	MODMIN	NEURAL	SAFDEC	NAG	DICE	MODMIN	NEURAL
louv	.191	.161	.181	.264	.445	.180	.205	.218	.352	.476	.105	.267	.290	.358	.528
walk	.091	.100	.121	.237	.428	.139	.098	.181	.349	.486	.174	.285	.334	.353	.526
gred	.249	.227	.245	.287	.471	.260	.209	.266	.365	.521	.315	.353	.372	.412	.541
inf	.204	.203	.220	.301	.383	.241	.259	.278	.407	.484	.255	.356	.358	.442	.561
lab	.124	.079	.162	.333	.374	.170	.203	.227	.428	.449	.155	.384	.412	.468	.541
RAND						.143				.185					.146
		Terrorists (terr)													
Deception Score		$\beta = 10\% C $				$\beta = 20\% C $				$\beta = 30\% C $					
		NAG	DICE	MODMIN	NEURAL	SAFDEC	NAG	DICE	MODMIN	NEURAL	SAFDEC	NAG	DICE	MODMIN	NEURAL
louv	.330	.200	.236	.293	.373	.359	.200	.402	.359	.452	.365	.317	.383	.411	.460
walk	.171	.239	.272	.360	.464	.252	.208	.287	.366	.475	.219	.155	.293	.382	.456
gred	.215	.208	.254	.289	.472	.231	.138	.200	.293	.382	.293	.276	.328	.424	.480
inf	.249	.274	.293	.260	.382	.214	.253	.335	.426	.446	.207	.289	.360	.395	.460
lab	.293	.322	.330	.367	.383	.281	.283	.293	.359	.413	.251	.293	.337	.379	.451
RAND						.115				.284					.224
		Terrorist attacks (terratt)													
Deception Score		$\beta = 10\% C $				$\beta = 20\% C $				$\beta = 30\% C $					
		NAG	DICE	MODMIN	NEURAL	SAFDEC	NAG	DICE	MODMIN	NEURAL	SAFDEC	NAG	DICE	MODMIN	NEURAL
louv	.102	.162	.242	.261	.352	.361	.279	.307	.484	.487	.351	.351	.445	.438	.539
walk	.192	.205	.253	.394	.544	.142	.101	.185	.357	.497	.312	.322	.358	.452	.563
gred	.271	.214	.343	.359	.376	.266	.214	.272	.374	.533	.351	.365	.491	.537	.556
inf	.359	.275	.350	.253	.554	.246	.265	.285	.417	.495	.349	.288	.361	.527	.593
lab	.404	.339	.437	.448	.576	.174	.207	.233	.437	.459	.445	.220	.420	.491	.606
RAND						.115				.163					.192

Fig. 2. Deception score for different values of budgets (β).

5 Community Deception: Where We Should Go

Despite the plethora of approaches to discover communities, it seems that there is not enough awareness that people can act strategically to evade such network analysis tools. This is particularly critical if who wants to *evade* such tools are *malevolent* users and who *run* the tools are *police* enforcement. This paper raises awareness on the fact that it is very reasonable to assume that terrorist groups, members of dark networks, and the like have their own set of deception tools to stay below the radar. There are a number of future research directions of interest for both sides of the barricade.

Other kinds of networks. Existing efforts have only focused on networks with no attributes and containing no edge labels. However, many real-world networks are endowed with attributes; for instance, social networks maintain additional information about the age, gender, and interests of users besides friendship relations. The challenge is to devise deception techniques that are able to counterbalance structure affinity (often summarized by the homophily principle) and attribute homogeneity considered by detection algorithms.

Embeddings. Besides traditional community detection techniques, there are a number of approaches that perform community discovery via (node and possibly edge) embeddings. Existing deception techniques are not suitable to work in such setting. The main challenge here consists in the fact that while in a non-embedding setting one can study the impact of edge updates on some optimization function (i.e., modularity minimization), understanding how updates reflect into the embedding space is not trivial.

Applying in deception practice. Algorithmic techniques need to be mapped into real world networks like Facebook or Twitter. The challenge here is how to turn community deception into a collective effort from \mathcal{C} 's member that, instructed by deception algorithms, rewire β updates according to a deception function ϕ . Note that while community detection algorithms require complete network knowledge, deception algorithms should ideally only need to know \mathcal{C} 's members and their links. In a network like Facebook, intra- \mathcal{C} (resp., inter- \mathcal{C}) edge deletions can be simply implemented by “Unfriending” some \mathcal{C} 's members (resp., external members). In Twitter, the same behavior can be achieved by “Unfollowing” some \mathcal{C} 's members (resp., external members). As for additions, in Facebook, which requires the acceptance of a friendship requests, an intra- \mathcal{C} edge addition would not represent a problem. Conversely, an inter- \mathcal{C} edge addition, which requires discovering new network members, can be implemented by picking the target node between colleagues, popular people, classmates, or even random people (by sending several friendship requests). In Twitter, this would reflect in just “Following” some member of the network. Understanding how to implement these policies “silently” is certainly challenging.

References

1. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Statist. Mech.* **2008**(10), 1–12 (2008)
2. Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M., Nikoloski, Z., Wagner, D.: On modularity clustering. *TKDE* **20**(2), 172–188 (2008)
3. Chakraborty, T., Srinivasan, S., Ganguly, N., Mukherjee, A., Bhowmick, S.: Permanence and community structure in complex networks. *ACM TKDD* **11**(2), 1–34 (2016)
4. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. *Phys. Rev. E.* **70**(6), 66–111 (2004)
5. Fionda, V., Palopoli, L., Panni, S., Rombo, S.E.: A technique to search for functional similarities in protein-protein interaction networks. *Int. J. Data Mining Bioinform.* **3**(4), 431–453 (2009)
6. Fionda, V., Pirrò, G.: Community deception or: how to stop fearing community detection algorithms. *IEEE Trans. Knowl. Data Eng.* **30**(4), 660–673 (2018)
7. Fortunato, S., Hric, D.: Community detection in networks: a user guide (2016). [arXiv:1608.00163](https://arxiv.org/abs/1608.00163)
8. King, G., Pan, J., Roberts, M.E.: How censorship in China allows government criticism but silences collective expression. *Am. Politic. Sci. Rev.* **107**, 326–343 (2013)
9. Liu, Y., Liu, J., Zhang, Z., Zhu, L., Li, A.: Rem: from structural entropy to community structure deception. *Adv. Neural Inf. Process. Syst.* **32**, 12938–12948 (2019)
10. Mittal, S., Sengupta, D., Chakraborty, T.: Hide and seek: outwitting community detection algorithms. *IEEE Trans. Comput. Soc. Syst.* (2021)
11. Nagaraja, S.: The impact of unlinkability on adversarial community detection: effects and countermeasures. In: PETS, pp. 253–272 (2010)
12. Newman, M.E.J.: Modularity and community structure in networks. *PNAS* **103**(23), 8577–8582 (2006)
13. Pons, P., Latapy, M.: Computing communities in large networks using random walks. In: ISCIS, pp. 284–293 (2005)
14. Nandini Raghavan, U., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E.* **76**(3), 036106 (2007)
15. Remy, C., Rym, B., Matthieu, L.: Tracking bitcoin users activity using community detection on a network of weak signals. In: Cherifi, C., Cherifi, H., Karsai, M., Musolesi, M. (eds.) *Complex Networks & Their Applications VI. COMPLEX NETWORKS 2017. Studies in Computational Intelligence*, **689**, 166–177. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-72150-7_14
16. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. *PNAS* **105**(4), 1118–1123 (2008)
17. Strogatz, S.H.: Exploring complex networks. *Nature* **410**(6825), 268–276 (2001)
18. Waniek, M., Michalak, T., Rahwan, T., Wooldridge, M.: Hiding individuals and communities in a social network (2016). [arXiv:1608.00375](https://arxiv.org/abs/1608.00375)



Mitigating the Backfire Effect Using Pacing and Leading

Qi Yang¹(✉), Khizar Qureshi¹, and Tauhid Zaman²

¹ Massachusetts Institute of Technology, Cambridge, MA 02139, USA
{yangqi,kqureshi}@mit.edu

² Yale University, New Haven, CT 06520, USA
tauhid.zaman@yale.edu

Abstract. Online social networks create echo-chambers where people are infrequently exposed to opposing opinions. Even if such exposure occurs, the persuasive effect may be minimal or nonexistent. Recent studies have shown that exposure to opposing opinions causes a backfire effect, where people become more steadfast in their original beliefs. We conducted a longitudinal field experiment on Twitter to test methods that mitigate the backfire effect while exposing people to opposing opinions. Our subjects were Twitter users with anti-immigration sentiment. The backfire effect was defined as an increase in the usage frequency of extreme anti-immigration language in the subjects' posts. We used automated Twitter accounts, or bots, to apply different treatments to the subjects. One bot posted only pro-immigration content, which we refer to as arguing. Another bot initially posted anti-immigration content, then gradually posted more pro-immigration content, which we refer to as pacing and leading. We also applied a contact treatment in conjunction with the messaging based methods, where the bots liked the subjects' posts. We found that the most effective treatment was a combination of pacing and leading with contact. The least effective treatment was arguing with contact. In fact, arguing with contact consistently showed a backfire effect relative to a control group. These findings have many limitations, but they still have important implications for the study of political polarization, the backfire effect, and persuasion in online social networks.

Keywords: Social networks · Social media · Political polarization · Computational social science

1 Introduction

Today online social networks provide a platform for one to persuade a potentially large audience [13]. However, the structure of these networks present their own obstacles to persuasion. Because users can choose from whom they receive information, these networks exhibit a great deal of homophily, where neighbors have similar opinions [3]. This creates echo-chambers where users are not frequently

exposed to arguments contrary to their own positions and existing opinions are often reinforced. Moreover, even if users are exposed to opposing views, empirical research has shown that when opinions differ greatly, making an argument can actually cause the opinions of the audience to shift away from the argument [2,9,11]. This *backfire effect* poses a major challenge when trying to persuade or influence individuals.

Within such online settings it has been found that the use of uncivil or extreme language can spread in such online settings [5]. Such language can create animosity among social media users and prevent constructive discussions. Given the scale and importance of online social networks, it is important to develop methods to persuade in these environments. However, the combination of the backfire effect and echo-chambers present major obstacles to persuasion. The structure of echo chambers prevent one from seeing contrary opinions, but if one does, the backfire effect limits their persuasion ability. It would be useful to have a method that allows one to present arguments in online social networks in a manner that mitigates the backfire effect and the usage of extreme language.

In this work we conduct a field experiment to test persuasion methods in an online social network. Our standard method, which we refer to as *arguing*, simply has one present arguments for the target position without any other interaction with the audience. Arguing can be viewed as a messaging based persuasion method because it only involves content posted by the arguer. We test another messaging method we refer to as *pacing and leading* which is based on the idea that persuasion is more effective if there is some sort of bond or connection between arguer and audience. This method begins by having the arguer emotionally pace the audience by agreeing with their opinion on the persuasion topic. This is done to form a bond with the audience. Then over time, the arguer shifts its own opinion towards the target position which will lead the audience to this position. In addition to messaging based methods, we also test a persuasion method based on interaction with the audience that we refer to as *contact*. This method has the arguer like the social media posts of its audience. This interaction can serve as a form of social contact in an online setting and potentially lead to more effective persuasion when combined with messaging based methods.

Our experiment tests two primary hypotheses. The first hypothesis is that pacing and leading will mitigate the backfire effect more than standard arguing through the effect of in-group membership, which means that the arguer and audience belong to a common social group. Theories of inter-group conflict suggest that persuasion is more effective when the arguer and audience are in-group [16]. In [10] race was used as an in-group feature to persuade users in the online social network Twitter to not use extreme language. It was found that in-group persuasion (arguer and audience have the same race) was more effective than out-group persuasion (arguer and audience have different races). This study demonstrated that race was an effective in-group feature for persuasion. We expect a similar finding when in-group membership is based on the opinion towards the persuasion topic.

Our second hypothesis is that contact between the arguer and audience will mitigate the backfire effect. By having contact with the audience, the arguer can form a rapport with the audience and shift them to a more positive affective state. Persuasion strength may be enhanced by these psychological effects. Researchers have found that affective states impact the efficacy of persuasion [14, 15]. The social influence literature is rife with evidence that social rapport and a positive relationship enhance persuasion and influence [6]. Moreover, it has been found that a person’s persuasive ability is strengthened if the audience likes this person [4].

2 Experiment Design

The persuasion topic used in our study is immigration. Events such as the European refugee crisis have made immigration a charged political issue and it is an active topic of discussion on social networks. Several studies have measured population level sentiment on this topic in Twitter [1, 7, 12]. It was found in [12] that English posts about the refugee crisis were more likely to have a negative opinion on the topic. A similar result was found for Twitter users in the United Kingdom [7]. Given the level of interest in the topic and its geo-political importance, immigration is an ideal topic to test persuasion methods. In our experiment we try to persuade individuals to have a more positive opinion of immigration.

We employ automated Twitter accounts, which we refer to as bots, to test different persuasion methods. Our experiment subjects are Twitter users who actively discuss immigration issues and have anti-immigration sentiment. Each bot implements a different persuasion method. One bot is a control which posts no content and does not interact with the subjects. One bot applies the arguing method by posting content which is pro-immigration. The third bot applies pacing and leading by posting content that is initially anti-immigration and then gradually become more pro-immigration. To test the contact treatment, we randomly selected half of the subjects from each bot and have the bots like the posts of these subjects. To assess the effectiveness of the different persuasion methods, we analyze the sentiment of content posted by these subjects over the course of the experiment. We now present details of our experiment design, which is illustrated in Fig. 1.

The subjects for our experiment were Twitter users who have an anti-immigration sentiment. To find potential subjects we began by constructing a list of phrases that conveyed strong anti-immigration sentiment, such as #CloseThe-Ports, #BanMuslim, and #RefugeesNotWelcome. We used the Twitter Search API to find posts, known as tweets in Twitter, that contained at least one of these keywords. We then collected the screen names of the users who posted these tweets.

Our search procedure has the potential to find users who do not have anti-immigration sentiment. For instance, to convey support for immigrants, a user could post a tweet critical of an anti-immigration phrase. To make sure that there were not many users who fall in this category, we manually investigated

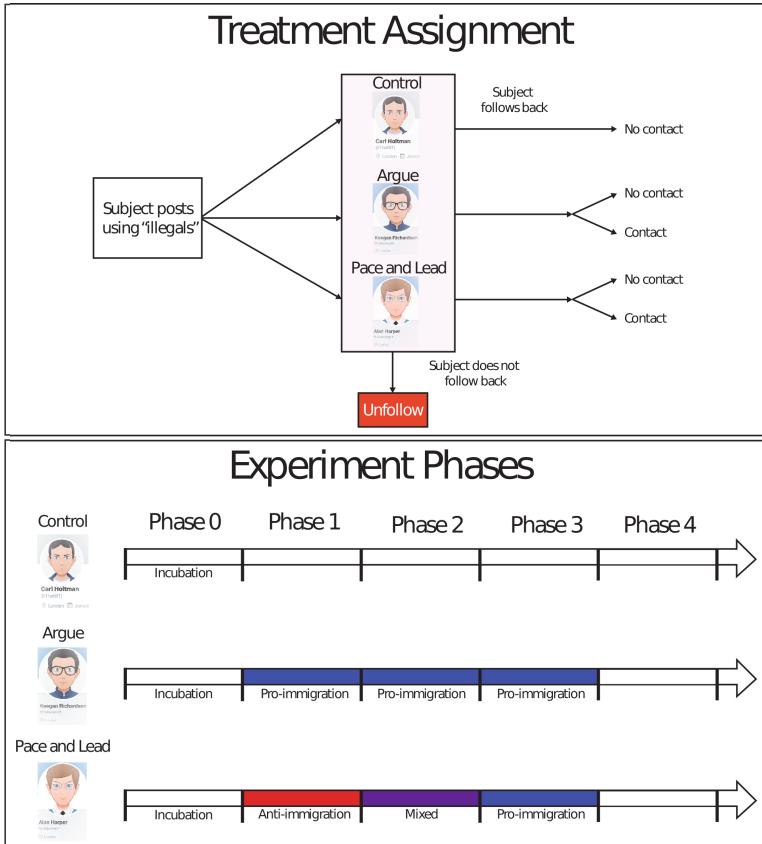


Fig. 1. (top) Diagram illustrating the subject acquisition procedure for the experiment. (bottom) Timeline of experiment phases.

100 random users collected by our search procedure. We found that none of the users was pro-immigration, giving us confidence that the overwhelming majority of our potential subjects were anti-immigration.

We further narrowed our subject pool by requiring each user to satisfy the following criteria. First, their tweet must be in English and must not contain only punctuation or emojis. Second, the user should not be an automated bot account. The text conditions on the tweet were checked using simple pattern matching. Bot accounts were identified using the machine learning based Botometer algorithm [8]. Users who Botometer identified as being the most bot-like were manually reviewed and eliminated if they are indeed bots.

We created Twitter accounts for the control, argue, and pace and lead treatments. One of the goals of our experiment was to test persuasion strategies in a realistic setting. Therefore, we wanted the bots to resemble human Twitter users, in contrast to the study in [2] where the subjects were told in advance the

Twitter account they were following was a bot. To accomplish this, we had the bots be active on Twitter for a two month incubation period before we started the experiment. Each of the bots location was set to London, and they followed a number of popular British Twitter accounts. The bots were designed to look like white males with traditional European names. We used cartoon avatars for the profile pictures, similar to what was done in [10]. We show the profile images for the bots and list their treatment type in Fig. 1. During the incubation period, once or twice a day the bots posted tweets about generic, non-immigration topics and shared tweets about trending topics on Twitter, an act known as retweeting. They also tweeted articles or videos talking about immigration, but not yet taking a stance on the issue. This was done to show that the bots had some interest in immigration before the experiment began.

One month into the incubation period, we began obtaining subjects for the experiment. To participate in the experiment, the potential subjects needed to follow the bots so that the bots' tweets would be visible in their Twitter timelines. We randomly assigned each of the users in the subject pool to the bots. The bots then liked a recent tweet of their assigned users and followed them. The liking of the tweet and following were done to increase the follow-back rate of the potential subjects. To avoid bias before the experiment, all tweets the bots liked were manually verified to not be immigration related. After liking and following their assigned subjects' tweets, the bots were able to achieve an average follow back rate of 19.3%. In total we were able to obtain 1,336 subjects who followed the bots. To make the bots appear more human, we tried to keep their ratio of followers to following greater than one. To do this, the bots would wait one to seven days before unfollowing a user who did not follow-back. The actual wait time depended on the user activity level, with a longer wait time given for less active users.

The experiment had four different phases. We denote the incubation period as phase zero. Phases one, two, and three are the main active phases of the experiment. The control bot does nothing for these phases. The argue bot would post a pro-immigration tweet once a day in these phases. The pace and lead bot also posted tweets once a day in these phases, but the tweet opinion varied. In phase one the tweets were anti-immigration. In phase two the tweets expressed uncertainty about immigration or potential validity of pro-immigration arguments. In phase three the tweets were pro-immigration, similar to the argue bot. We constructed the tweets based on what we deemed a proper representation of the opinion for each phase. In phase four of the experiment the bots tweeted nothing. We used this phase to measure any persistent effect of the treatments. Each phase lasted approximately one month, except for the incubation phase which lasted two months. The incubation phase began on September 27th, 2018 and the fourth phase was completed on March 1st, 2019. The experiment timeline is shown in Fig. 1.

In addition to the tweeting based treatments, we also tested the contact treatment on the subjects. We randomly assigned 50% of the subjects of the argue and pace and lead bots to this treatment group. During phases one, two, and three, the bots liked the tweets of the subjects assigned this treatment. When the bot liked a subject's tweet, the subject is notified. Liking tweets would make the bot more visible to the subject and potentially give the subject a greater trust or affinity for the bot. The control bot did not apply the contact treatment to any of its subjects.

All subjects voluntarily chose to follow the bots, which may lead to a selection bias in our subjects. Therefore, our conclusions are limited to Twitter users willing to follow the bots and do not necessarily generalize to all Twitter users. However, since a follow-back is required for a Twitter account to implement a tweet based treatment, this is not a strong limitation of our conclusions. This experiment was approved by the Institutional Review Board (IRB)¹ for the authors' institution and performed in accordance with relevant guidelines and regulations.

3 Results

We used the frequency of extreme anti-immigration language in the subjects' tweets to measure any persuasion effect the bots had. In particular, we counted how many of the subjects' tweets contained the word "illegals" in each phase. The term illegals is a pejorative term used by people with anti-immigration sentiment. For instance, there are tweets such as *I want a refund on all the tax money spent on illegals!!!* which show strong anti-immigration sentiment. The usage frequency of such extreme language can be used to gauge sentiment, as was done in [10]. We chose the word illegals because it is consistently used by anti-immigration Twitter users, unlike hashtags that gain temporary popularity. We plot the illegals usage frequency in each phase and treatment group in Fig. 2. This frequency is defined as the number of tweets containing illegals divided by the total number of tweets for all subjects in each phase and treatment group. We note that the overall frequency is very low, but shows aggregate differences between phases. For instance, phase three has a higher frequency than the other phases for all treatments. This suggests that there are exogenous factors affecting the behavior of the subjects. Another interesting observation is in phase two, where we see that the pace and lead with contact treatment has a much lower frequency than the other treatments, while argue with contact has the highest frequency. Recall that in phase two pacing and leading has tweets that are slightly pro-immigration. We next perform a more quantitative statistical analysis to assess the different treatments.

¹ The protocol has been approved following full board review by the Committee on the Use of Humans as Experimental Subject (COUHES) under protocol number 1808496544.

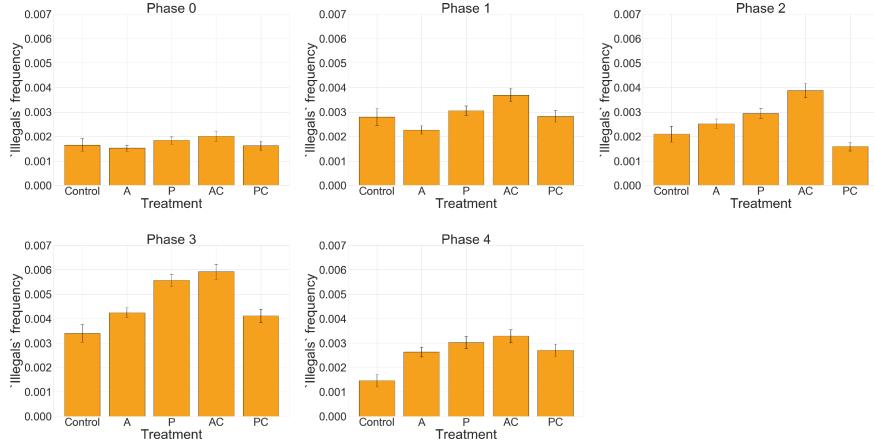


Fig. 2. Plot of the frequency and standard error of usage of the word “illegals” in tweets for each phase and treatment group. The treatments are labeled as follows: A is argue without contact, AC is argue with contact, P is pace and lead without contact, and PC is pace and lead with contact.

We treat each tweet as a binary outcome that equals one if the tweet contains the word illegals. The probability of such an outcome is modeled using logistic regression. For a tweet i the probability is

$$\log \left(\frac{p_i}{1 - p_i} \right) = \sum_{t=0}^4 \beta_t x_{t,i} + \sum_{t=0}^4 \beta_{a,t} x_{a,i} + \sum_{t=0}^4 \beta_{p,t} x_{p,i} + \sum_{t=0}^4 \beta_{ac,t} x_{ac,i} + \sum_{t=0}^4 \beta_{pc,t} x_{pc,i} + \epsilon_i. \quad (1)$$

The coefficients β_t for $t = 0, 1, \dots, 4$ model exogenous factors that may impact the probability during each phase. For instance, news stories related to immigration may increase the probability. We use separate treatment coefficients for each phase because the pace and lead treatment varies by phase. Recall that this treatment shifts the opinion of its tweets from anti- to pro-immigration over phases one to three. The treatment coefficients are indexed by subscripts indicating the treatment and phase. We use the subscript t for the phase, a for argue, and p for pace and lead. The subscript c indicates the contact treatment where the bots like the subjects’ tweets. The x variables are binary indicators for the treatment group of the subject posting the tweet and in which phase the tweet occurred. User heterogeneity and other unobserved factors are modeled using a zero mean normally distributed random effect ϵ_i .

By regressing out the phase effect we can isolate the different treatments. We plot the resulting treatment coefficients separated by tweet group (argue or pace and lead) and contact group in Fig. 3. This grouping makes differences in each individual treatment over the phases more visible. We also indicate on the plots which differences are statistically significant at a 1% level.

We first look at the effect of the contact treatment. In the top left plot of Fig. 3 we see that the argue with contact coefficient is greater than argue without contact, and the difference does not vary much over the phases. The difference is significant for phases one, two, and three. In phases zero and four, where the bots do not tweet about immigration, there is no significant difference. The contact treatment may be making the bots' pro-immigration tweets more visible to the subject, resulting in a backfire effect where the subject uses the word illegals more frequently.

For pacing and leading in the top right plot of Fig. 3, we see that the non-contact coefficient is greater than contact. In phases two and three the difference is significant. Contact appears to enhance the effectiveness of pro-immigration tweets in the later stages of the pacing and leading treatment. This is in contrast to arguing, where contact degrades the effectiveness of pro-immigration tweets.

We next look more closely at arguing versus pacing and leading when the contact treatment is fixed. In the bottom left plot of Fig. 3 we see that without contact, the tweet treatment coefficients have a small difference which does vary appreciably across phases. Argue has a smaller coefficient, but the difference is statistically significant only for phases one and three.

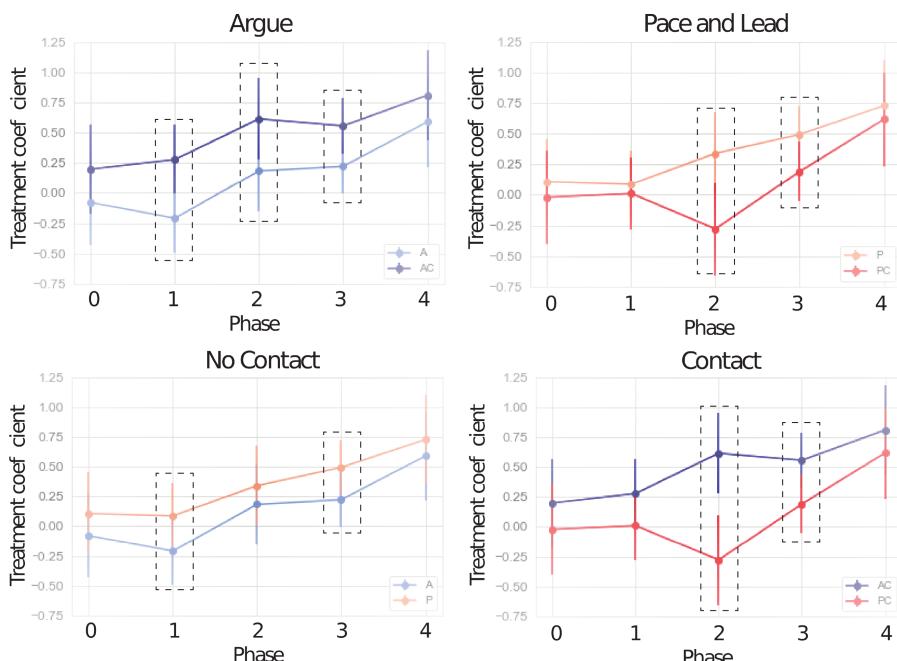


Fig. 3. Plots of the regression coefficients (with standard errors) for the treatments in each phase. The title of each plot indicates the treatment component that is held fixed. The treatments are labeled as follows: A is argue without contact, AC is argue with contact, P is pace and lead without contact, and PC is pace and lead with contact. The dashed boxes indicate which coefficients have a difference that is statistically significant at a 1% level.

For the contact group in the bottom right plot of Fig. 3, the difference changes sign. Argue has the larger coefficient and the difference varies across the phases. Phase two shows a large significant difference. The difference is smaller in phase three, but still significant. The moderately pro-immigration tweets of the phase two pace and lead treatment seem to be more effective than the argue tweets when the bot has contact with the subject. The same can be said of fully pro-immigration tweets in phase three, but the advantage of pacing and leading over arguing is less than in phase two.

4 Discussion and Conclusion

Our results show that when the bots make contact with the subjects, pacing and leading was more effective than arguing in phases two and three. If the bots were arguing, then contact had the opposite effect and made the treatment less effective. We see a novel interaction effect, where combining pacing and leading with contact is the most effective treatment, especially in phase two.

Our findings suggest strategies one can use to overcome the challenges posed by echo-chambers and the backfire effect. We were able to penetrate echo-chambers by using bots which followed and liked the posts of the users. This follow and like method proved to be quite effective at getting users to follow the bots. Penetrating an echo-chamber allows one to present arguments to the user. To overcome the backfire effect, we found that the bots should continuously like the posts of the users, and present arguments that are more nuanced and moderate in their language (phase two of the pacing and leading treatment). This softer approach proved more effective than standard arguing.

There are several interesting questions raised by our findings. One question concerns the phases for pacing and leading. We found that the moderate posts were most effective. It is not clear if this treatment would work in isolation or if the phase one pacing treatment is necessary. We hypothesize that this period allows greater trust to be built between subject and bot, but our experiment does not confirm this. Another question is whether the phase three pace and lead treatment where the posts strongly advocate the target position is necessary. It may be that the moderate posts are sufficient to mitigate the backfire effect and potentially persuade the subject.

Finally, we note that care should be taken when trying to apply our results to more general settings. This study focused on the topic of immigration, which is an important political and policy issue. Discussion on this topic has split along traditional conservative liberal fault lines. We expect our findings to extend to similar political issues, but further study is needed. However, our subjects were Twitter users with anti-immigration sentiment who were willing to follow our bots. This represents a limited population in a very specific social setting. More work is needed to determine whether our findings replicate in different populations or within varied social settings.

References

1. Backfried, G., Shalunts, G.: Sentiment analysis of media in German on the refugee crisis in Europe. In: International Conference on Information Systems for Crisis Response and Management in Mediterranean Countries, pp. 234–241. Springer, Cham (2016)
2. Bail, C.A., et al.: Exposure to opposing views on social media can increase political polarization. *Proc. Natl. Acad. Sci.* **115**(37), 9216–9221 (2018)
3. Bakshy, E., Messing, S., Adamic, L.A.: Exposure to ideologically diverse news and opinion on Facebook. *Science* **348**(6239), 1130–1132 (2015)
4. Burger, J.M., Soroka, S., Gonzago, K., Murphy, E., Somervell, E.: The effect of fleeting attraction on compliance to requests. *Personal. Soc. Psychol. Bull.* **27**(12), 1578–1586 (2001)
5. Cheng, J., Bernstein, M., Danescu-Niculescu-Mizil, C., Leskovec, J.: Anyone can become a troll: Causes of trolling behavior in online discussions. In: Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, pp. 1217–1230 (2017)
6. Cialdini, R.B., Trost, M.R.: Social influence: social norms, conformity and compliance(1998)
7. Coletto, M., et al.: Sentiment-enhanced multidimensional analysis of online social networks: perception of the Mediterranean refugees crisis. In: Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 1270–1277. IEEE Press (2016)
8. Davis, C.A., Varol, O., Ferrara, E., Flammini, A., Menczer, F.: Botornot: a system to evaluate social bots. In: Proceedings of the 25th International Conference Companion on World Wide Web, pp. 273–274. MISSING PUBLISHER (2016)
9. Lord, C.G., Ross, L., Lepper, M.R.: Biased assimilation and attitude polarization: the effects of prior theories on subsequently considered evidence. *J. Personal. Soc. Psychol.* **37**(11), 2098 (1979)
10. Munger, K.: Tweetment effects on the tweeted: experimentally reducing racist harassment. *Polit. Behav.* **39**(3), 629–649 (2017)
11. Nyhan, B., Reifler, J.: When corrections fail: the persistence of political misperceptions. *Polit. Behav.* **32**(2), 303–330 (2010)
12. Öztürk, N., Ayvaz, S.: Sentiment analysis on twitter: a text mining approach to the Syrian refugee crisis. *Telemat. Inform.* **35**(1), 136–147 (2018)
13. Perrin, A.: Social media usage. Pew Research Center, pp. 52–68 (2015)
14. Rind, B.: Effects of interest arousal on compliance with a request for help. *Basic Appl. Soc. Psychol.* **19**(1), 49–59 (1997)
15. Rind, B., Strohmetz, D.: Effect on restaurant tipping of presenting customers with an interesting task and of reciprocity. *J. Appl. Soc. Psychol.* **31**(7), 1379–1384 (2001)
16. Tajfel, H., Turner, J.C., Austin, W.G., Worchsel, S.: An integrative theory of inter-group conflict. *Organiz. Ident. Read.* **56**, 65 (1979)



Exploring Bias and Information Bubbles in YouTube's Video Recommendation Networks

Baris Kirdemir^(✉) and Nitin Agarwal

Cosmos Research Center, U.A. – Little Rock, Little Rock, AR, USA

Abstract. This study audits the structural and emergent properties of YouTube's video recommendations, with an emphasis on the black-box evaluation of recommender bias, confinement, and formation of information bubbles in the form of content communities. Adopting complex networks and graphical probabilistic approaches, the results of our analysis of 6,068,057 video recommendations made by the YouTube algorithm reveals strong indicators of recommendation bias leading to the formation of closely-knit and confined content communities. Also, our qualitative and quantitative exploration of the prominent content and discourse in each community further uncovered the formation of narrative-specific clusters made by the recommender system we examined.

Keywords: Bias · Recommendation algorithm · k-core · Modularity · Information bubble

1 Introduction

Detection, understanding, prediction, characterization, and mitigation of bias in modern systems of automated and autonomous decisions is a growing interdisciplinary field. This study focuses on the recommender systems employed by online social networks as one of the major components of the given domain. Problems relating to bias, unfairness, insufficient coverage, lack of diversity, transparency, or explainability of recommendations across online social networks may have significant societal implications. The recent literature suggests that popularity bias arises when recommender systems implicitly or explicitly boost the popularity of already popular items further, showing a characteristic behavioral pattern observed in a variety of applications [1]. In addition, recommender systems within the given context may exacerbate problems relating to the echo chambers, polarization, and information bubbles in which alternative facts, conspiracy theories, and false information thrive at the expense of inter-community communication and continual exchange of ideas. Recent studies documented that implicit bias in recommender systems range from the gender bias in Facebook's job ads [2] and promotion of health misinformation in Amazon's book recommendations [3] to potential bias in recommendations that may boost only hyperactive users' influence across online social networks [4].

This study aims to explore YouTube's video recommendation bias and emergent clusters of content communities. As one of the most popular social media platforms, YouTube's search, ranking, personalization, and recommendation systems have been under journalistic and academic scrutiny in recent years. Several accounts suggest that, despite the multitude of countermeasures officially announced by the platform, YouTube's algorithms boost echo chambers, polarization, as well as extremist and harmful content in certain conditions [5, 6]. In particular, users' entry points to the platform and video recommendations are suggested as potential conditions that lead to the formation of confined and closely-knit clusters, strengthening the influence and echo-chamber effects of ideological, extremist, and polarizing content [5]. However, others have suggested that the evidence of bias towards extremist channels is lacking [7, 8], emphasizing the fact that the number of systematic, scholarly, and large-scale examinations of the given phenomenon is minimal, and the academic literature lags behind the public debate partly because of the complex and black-box characteristics of YouTube's recommendation and personalization systems.

The rest of the paper is organized as follows. Section 2 provides a literature survey. Section 3 describes data collection and research methods. Section 4 presents key findings and analysis. Section 5 concludes the paper with future directions.

2 Literature Review

This section will introduce the relevant literature broken into three themes, viz., recommender bias in general, recommender bias in YouTube in particular, and analysis of recommendation networks.

Recommender Bias. Search, ranking, and recommendation systems rely on various system architectures, historical data, user data, personalization algorithms, and a variety of other information sources [13]. Surveying 22 recent papers in the given field, Verma et al. [14] identified five different categories of recommender settings, including non-personalized recommendation, crowd-sourced recommendation, personalized recommendation, online advertisements, and marketplace, while also distinguishing between diversity, novelty, and relevance as the main aspects of such systems. Popularity bias, relating to the problem of superstar economics [15], has been one of the most extensively studied types of recommender bias, as in many real-world examples the accuracy, diversity, exposure, and coverage of recommendations are affected when the popular items become more popular due to feedback loops [1]. Popularity bias affects a wide variety of user-item-provider marketplaces ranging from music recommendations [16] where the exposure of niche artists suffer, to platforms of massive online courses where it also strengthens coverage and concentration biases of ranked courses [17]. Adding to the item or provider popularity, user or provider hyperactivity may alter the distribution of recommendations and lead to biased coverage of information [4]. Olteanu et al. [18] identifies four types of bias in relation to platforms, viz., functional bias, normative bias, external bias, and bias emanating from non-individual accounts. Bartley et al. [19] demonstrated that Twitter's recommendations produce skewed

information distributions, negatively affecting novelty, fairness, and exposure in terms of how frequently users appear on others' timelines.

Bias in YouTube Recommendations. In recent years, concerns about the idea of harmful content leading to the formation of echo chambers, polarization, and radicalization on social media platforms have been shared by many [5, 7, 20–22]. Motivated by their suspicion of the filter bubble effect, Faddoul et al. [20] developed a classifier in an attempt to automatically categorize YouTube's video recommendations as conspiratorial. Hussein et al. [21] found that demographics such as geolocation, age, and gender affect the level of misinformation content in recommendations, but only in cases where a user has developed a viewing history and not for brand new accounts. The researchers analyzed non-political misinformation topics and found that YouTube's recommendation algorithm pushed specific misinformation themes like chemtrails, flat earth, and the faked moon landing. Both groups of researchers [20, 21] have found that there is a clear correlation between the videos that were watched and the videos that were recommended, i.e., both were of conspiratorial nature, creating the filter-bubble effect. Roth et al. [22] further analyzed filter-bubbles or topological confinement. The researchers concluded that these bubbles are focused around most watched videos.

Ledwich and Zaitsev [7] analyzed YouTube's role in online radicalization. The study concluded that YouTube does not promote radical or far-right content at all and, instead, consistently leads users towards moderately left or mainstream channels. On the contrary, Ribeiro et al. [5] suggested that users are exposed to increasingly extreme content only if they had previously viewed other videos associated with radicalization. These studies dive deeper into the filter bubble effect and clearly demonstrate the need for a comprehensive look at YouTube recommendations in different contexts and over long time periods. The work presented in this paper is an attempt towards that direction.

Analysis of Recommendation Network. The use of network analysis approaches can aid in examining the latent characteristics of recommendation algorithm and their impact on users. Graphs are used to represent nodes that could be videos, channels, pre-defined content categories, or inferred classes of videos, and edges are the recommendations. The basic topology of the recommendation graph is used in the operationalization of concepts such as bias, fairness, confinement, filter bubbles, echo chambers, polarization, and radicalization. This is the case in both node-centric [22] and user-centric (personalization) [21] research. The data is often captured through multiple hops of graph crawling.

Principles of complex networks, particularly PageRank inspired network centrality measures, have been utilized to study characteristics of recommendation algorithms, whether it is Amazon's book recommendation algorithm [3], music recommendation algorithms [12], YouTube's video recommendation algorithm [10, 11], movie recommendations [23], or recommendations in general [24]. These studies, irrespective of the application domain, reported that recommendations followed a power-law distribution, a characteristic of small-world networks that have a long/heavy tail. This implies that few items are recommended the most (the short head), while most items are highly infrequently recommended (i.e., the long tail) that often feeds to the popularity bias. Furthermore, Le Merrer and Trédan [9], through network analysis of successive

recommendations, examined the detection of recommendation biases that are explicitly induced by service providers.

The application of complex networks' principles has shed light on the properties of recommendation graphs. Mirza et al. [23] introduced the term hammock jump that identifies the minimum number of ratings or "jumps" needed to keep the recommendation graph connected, shrinking the resultant average path length between users. Further, by introducing biased edges in the recommendation graph, one could achieve a more compact average path length [9] - a finding that helps recommendation algorithms deal with sparse networks. In this work, we show that an extended stochastic analysis of the video recommendation network from YouTube leads to a better understanding and characterization of its underlying bias that may have a significant influence on the dissemination, platforms, consumers, and providers of information. The following section describes the dataset and methods we used in our experiments.

3 Data and Methods

We approach the given problem from complex networks and graphical probabilistic perspectives. Extending from the relevant literature [9–12], we posit that employing topological network analysis measures and a stochastic approach to examine YouTube's video recommendations provide a strong base for exploring, understanding, and characterizing the structural properties of the video recommendation networks. With the evaluation of the structural and emergent features of YouTube's video recommendation networks, we aim to contribute to the given research domain related to implicit recommender bias, emergent information bubbles, content communities, and complex networks. To accomplish this goal, we systematically studied 6,068,057 recommendations made by the YouTube Algorithm.

We utilized the VTracker tool [25] and the methodology described in Kready et al. [26] for data collection, using the YouTube Data API [27]. First, we acquired a large set of seed videos from 6 distinct influencer channels that host content widely consumed by online alt-right communities, found by following a slightly modified process described in [5]. The original publishing dates range from January 5, 2009, to June 20, 2021, amounting to 5,114 distinct records in the collection of seed videos. Next, we recursively collected data of related videos (recommendations) that were recommended to the viewers of each video. In total, the recommended videos dataset included 6,068,057 records with a timeframe from January 9, 2006, to June 19, 2021, based on video publish dates. To reduce the number of recommendations for each parent video, we cleaned the redundant and duplicated observations. As a result, each recommended video had only one incoming link from a single parent video in the final dataset. In addition, we collected available comments made for each recommended video (if there are any). The total number of comments amounted to 10,002,035. As described in the following sections, we recursively combined the video titles, video descriptions, and comments to assess the content-based communities and compare the clusters emerging from the collected set of recommendations.

Using the video recommendations dataset described above, we first built the full recommendation network by drawing edges between each "parent video" and

corresponding “recommended videos”. We then utilized the methodology described in Kirdemir et al. [10, 11], using a node-specific probabilistic approach to detect the items that gain influence over the rest of the graph and can be visited more frequently by users as a result of the recommendations made by the given algorithm. Namely, we extracted each node’s PageRank [28] score in the recommendation graph to examine each video’s probability of being visited in a random walk scenario.

Second, we explored the intermediate level organization, sub-graphs, and formation of content communities in the recommendation network extracted from the empirical data. As explained in the next section, we also aggregated the PageRank scores for each modularity class and channel in the recommendation graph for a group-level characterization of the network. Combined with the PageRank-based influence scores, the formation of content communities may indicate the emergent clusters and potential echo-chamber effects partially led by the recommendation algorithm. To this end, we computed the modularity in the recommendation graph and extracted modularity classes by using the Louvain method [29] and introducing a resolution parameter [30] to the process. The resolution parameter correlates with the number of content communities (modules) we find using the Louvain method. We introduce a conservative resolution parameter (3.50) to examine the content communities as described by [32] and visualize the clusters of recommended items in the network. Given the higher-than-average resolution we chose, the modules we extract from the network could be in intermediary orders in a hierarchical community-finding setting where overlapping communities are present. In addition, we explored influential channels and videos within each community by using the PageRank values gained by each node.

Last, we analyzed the recommendation graph by exploring the k-core structures to identify closely-knit sub-communities and potential confinement of recommendations or consumption of recommended content. Studies have shown that in real-world networks, the community structures and node-based or group-based influence may be interrelated with the k-core decomposition of the network [31]. The k-core structure (decomposition, degeneracy) of a network refers to the “*maximal set of nodes that have at least k neighbors*” [31] in the resulting subgraph [32, 33]. This study’s list of k-core subgraphs includes 10-core, 20-core, 30-core, 40-core, and 50-core structures. In each decomposition, we examine the change of the recommendation network and content communities as the weaker and less influential nodes are recursively filtered out. Furthermore, the interconnections between communities are explored across the k-core decompositions. Finally, the k-core analysis provides an additional perspective on whether the structure and content in each decomposition signal the presence of closely-knit sub-communities and a potential confinement of video recommendations.

4 Analysis and Results

This section examines the recommendation graph consisting of the videos as nodes and recommendations as edges. We also demonstrate the distribution of PageRank scores and whether a small number of videos, channels, or communities gain influence and dominate the rest of the graph in a random walk setting as a proxy of potential bias emerging from the recommendations. Next, we examine content communities

(modularity classes) based on the Louvain algorithm and influential channels within the communities by aggregating the node-based PageRank scores to the level of channels (content producers) for each community. This section also includes a discussion of the content-based comparison between the communities we extracted from the recommendation graph and explores the k-core decompositions described in the previous section.

Recommendation Network. In total, the recommendation graph includes 53,293 nodes (videos) and 243,914 edges (recommendations). The Louvain method finds ten different content communities with individual sizes ranging from 0.03% to 36.84% of the entire network. Four of the ten recommended video communities are larger than 10% of the recommendation graph. In the recommendation network, the distribution of node-specific PageRank scores is a long-tail distribution, similar to the findings of the previous studies using a similar approach. Also, a small fraction of nodes gain higher PageRank scores and influence in each content community.

A similarly skewed distribution is observed between the channel-level recommendations in the network, based on the total number of recommendations received, the sum of the PageRank scores, and the average, median, maximum, and minimum scores acquired by the videos belonging to each channel. Figure 1 shows the recommended channels in terms of the sum and mean of the PageRank scores and the total number of recommendations each channel received in a log-log plot. We note that five of the six most recommended channels are from the seed list of 6 channels we used for data collection. We also observe that the most influential channels also receive a vast majority of recommendations, given the total number of records we collected was 6,068,057.

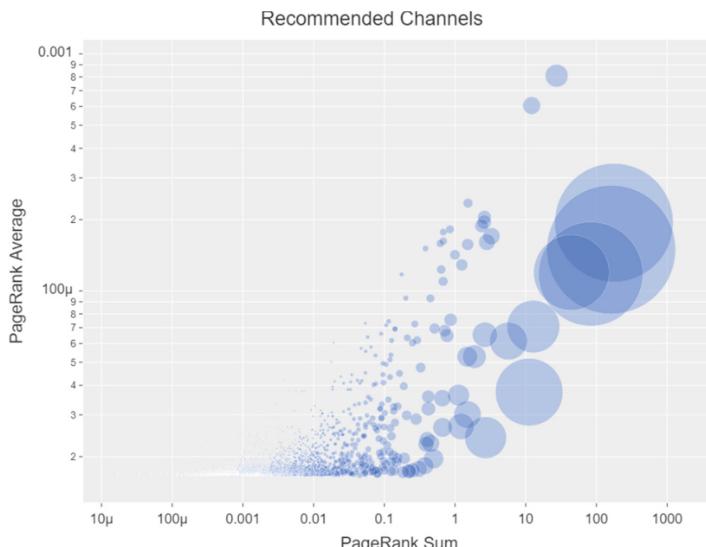


Fig. 1. The channels in the recommendation network. The log-log plot shows the sum of the PageRank scores in the x axis and mean P.R. scores in the y axis. The node size represents the number of recommendations received by channels.

In addition, the distribution of the mean PageRank values in the channel level demonstrates significant differences between the small fraction of highly influential channels and the rest of the graph. Overall, the average PageRank score for all channels was 0.00001749, while the most influential channel, in terms of the average scores, reached 0.000809. The greatest difference between the influential channels and the rest appears to be represented in the distribution of the sum of the PageRank scores. Accordingly, the average channel-level sum was 0.0388, while the five highest scoring channels gained 177.008, 161.850, 83.269, 44.335, and 27.351 in terms of the PageRank values.

Content Communities. We observed that the formation of content communities had several apparent characteristics, mostly in terms of influential channels, the frequency of recommendations each channel received, and the prominent types of content or narratives presented. Table 1 presents each modularity class (content community) with its size and distribution-related parameters. Some communities appear to have higher numbers of influential videos recommended by the YouTube algorithm. The number of recommendations each community received ranged from 137 to 2,243,485, with an average of 606,806.

Similarly, the top-scoring community gained 50 times higher average than all communities in terms of the mean PageRank scores. In addition, Table 1 includes the Maximum Likelihood Estimate of Alpha value for channel-level recommendation distributions for each content community. Larger communities have power-law (long-tail) characteristics internally.

Table 1. Description of content communities and channel-level recommendations for each community

Community	Size (Video recommendations)	Maximum likelihood estimate of alpha [34] (Channel-level recommendations)	Max. number of recommendations for a single channel	Most recommended channel
1	227,922	2.25	181,039	John Doyle
2	626,103	2.30	370,882	No White Guilt
3	1,458,997	2.00	1,082,898	StevenCrowder
4	1,507,273	1.95	706,052	Matt Walsh
5	137	4.73	23	Klayton Floriti
6	312	3.13	37	DaBaby
7	1236	3.13	154	The Enlightened Target
8	1257	3.57	318	Hitler Rants Parodies
9	2,243,485	1.88	911,125	Ben Shapiro
10	1335	3.54	94	Optimus

We repeatedly observed that most of the recommendations were received by the videos in a few modules. We note that some channels appeared in more than one

community, meaning that individual connections received by those channels were recommended in multiple clusters. Also, almost all the larger content communities were dominated by one or a few channels both in terms of the number of recommendations and mean PageRank scores achieved within each community. All in all, we find that the YouTube recommendation system tends to produce closely-knit clusters of recommended content that depend on channels being produced as the entry points to the recommender system.

Community-Level Assessment of Prominent Content and k-core Decompositions. Using the video titles, descriptions, and comments made for each video in our final dataset of YouTube's recommendations, we observed that each module in the recommendation graph has distinct, prominent topics and narratives presented to platform users. The prominence of narratives also seems to be related to a few channels' influence in each cluster, as presented in the previous subsection. Community 9, as the largest module with a large number of high PageRank scoring videos, is dominated by the narratives about the criticism of the "Black Lives Matter" movement, the mainstream media, liberal and progressive political movements, and the Biden Administration. Apparently, as part of a community-specific discourse used by influential channels, criticism of "systemic racism theory" frequently appears in the same content community. In community 2, the terms "white guilt" and "white erasure", and "white well-being" are the most frequent terms used in the given video attributes. In community 1, we observed the most influential terms related to narratives of anti-liberal, anti-communist, anti-Antifa, anti-gun-control characteristics. Overall, all communities we extracted from the recommendation network using the Louvain algorithm have distinct narratives with little overlap, and usually in relation to highly polarizing discourse and extremist ideologies in the contemporary political spectrum of the United States.

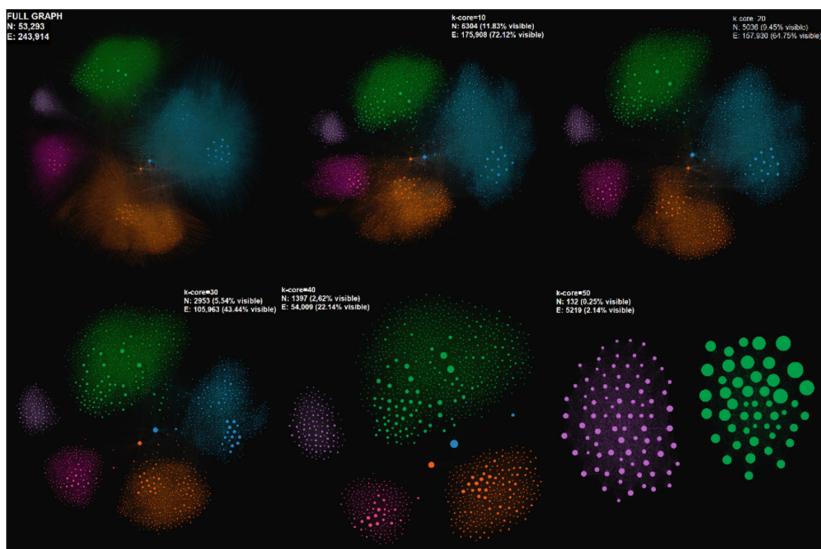


Fig. 2. k-core decompositions of the recommendation network. Colors represent the modules, while node size represents the PageRank scores gained in the entire graph.

In the final step of our analysis, we explored the k-core decompositions of the recommendation graph in five different levels ranging from 10-core to 50-core graphs. As demonstrated in Fig. 2, k-core decompositions reveal that the communities are highly closely-knit, meaning the recommendations have very little to no content diversity, giving rise to information bubbles. In the 10-core decomposition of the graph, only 11.83% of the nodes and 72.12% percent of the edges remain visible, and the most influential nodes continue to be present in the consecutive maximal subgraphs. As demonstrated, the largest module (community 9) tends to disappear as we increase the level of decomposition. In addition, k-core structures also show the level of influence each node gains in a community versus the entire graph. As the 30-core, 40-core, and 50-core subgraphs indicate, a set of highly influential nodes appear to be limited to their corresponding clusters as the primary source and area of their influence, and the graph becomes more confined and polarized as the stronger nodes remain as survivors.

Future analyses of the k-core structures in video recommendation networks may provide further insights into the survivability of closely-knit subgraphs in case of perturbations. In our case, such perturbations would include changing the recommendation algorithm, removing the harmful content detected by YouTube's internal safety systems, or information operations conducted by external actors to boost or diminish the present content producers on the platform.

5 Conclusions and Future Work

In this study, we consecutively used the PageRank algorithm, Louvain modularity method, k-core decompositions, and content analysis to explore the structure and emergent properties of YouTube's video recommendation networks. Our quantitative and qualitative exploration of the prominent content and discourse in each community further uncovered the formation of narrative-specific clusters made by the recommender system we examined. As demonstrated in the previous section, the use of network analysis in evaluating recommender systems and recommender bias provides a strong foundation for the development of models for auditing, predicting, understanding, and mitigating the effects of biased and unfair recommendations across online social networks. In future work, additional and repetitive hops of recursive data collection may enable further insights and introduce the longitudinal dimension. Also, alternative community detection approaches may enable greater granularity and uncover additional hidden properties in the recommendation networks. Also, future work would tie the complex networks approach to metrics of recommender system evaluation, such as diversity, coverage, utility, transparency, fairness, security, and exposure.

Acknowledgments. This research is funded in part by the U.S. National Science Foundation (OIA- 1946391, OIA-1920920, IIS-1636933, ACI-1429160, and IIS-1110868), U.S. Office of Naval Research (N00014-10-1-0091, N00014-14-1-0489, N00014-15-P-1187, N00014-16-1-2016, N00014-16-1-2412, N00014-17-1-2675, N00014-17-1-2605, N68335-19-C-0359,

N00014-19-1-2336, N68335-20-C-0540, N00014-21-1-2121), U.S. Air Force Research Lab, U. S. Army Research Office (W911NF-17-S-0002, W911NF-16-1-0189), U.S. Defense Advanced Research Projects Agency (W31P4Q- 17-C-0059), Arkansas Research Alliance, the Jerry L. Maulden/Entergy Endowment at the University of Arkansas at Little Rock, and the Australian Department of Defense Strategic Policy Grants Program (SPGP) (award number: 2020-106-094). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations. The researchers gratefully acknowledge the support.

References

1. Abdollahpouri, H., Burke, R., Mobasher, B.: Controlling popularity bias in learning-to-rank recommendation. In: Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys 2017) (2020)
2. Imana, B., Korolova, A., Heidemann, J.: Auditing for discrimination in algorithms delivering job ads. In: Proceedings of the Web Conference 2021 (WWW 2021), Ljubljana, Slovenia. ACM (2021). <https://doi.org/10.1145/3442381.3450077>
3. Juneja, P., Mitra, T.: Auditing e-commerce platforms for algorithmically curated vaccine misinformation. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI 2021). Association for Computing Machinery (2021). <https://doi.org/10.1145/3411764.3445250>
4. Papakyriakopoulos, O., Serrano, J.C.M., Hegelich, S.: Political communication on social media: a tale of hyperactive users and bias in recommender systems. Online Soc. Netw. Media **15**, 100058 (2020)
5. Ribeiro, M.H., Ottoni, R., West, R., Almeida, V.A., Meira Jr, W.: Auditing radicalization pathways on YouTube. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 131–141 (2020)
6. Tufekci, Z.: YouTube, the great radicalizer. The New York Times, 10 May 2018
7. Ledwich, M., Zaitsev, A.: Algorithmic extremism: examining YouTube’s rabbit hole of radicalization. First Monday (2020)
8. HosseiniMardi, H., Ghasemian, A., Clauset, A., Mobius, M., Rothschild, D. M., Watts, D.J.: Examining the consumption of radical content on YouTube. Proc. Natl. Acad. Sci. 118(32) (2021)
9. Le Merrer, E., Trédan, G.: The topological face of recommendation. In: Cherifi, C., Cherifi, H., Karsai, M., Musolesi, M. (eds.) Complex Networks & Their Applications VI, vol. 689, pp. 897–908. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-72150-7_72
10. Kirdemir, B., Kready, J., Mead, E., Hussain, M.N., Agarwal, N.: Examining video recommendation bias on YouTube. In: Boratto, L., Faralli, S., Marras, M., Stilo, G. (eds.) Advances in Bias and Fairness in Information Retrieval. CCIS, vol. 1418, pp. 106–116. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-78818-6_10
11. Kirdemir, B., Kready, J., Mead, E., Hussain, M.N., Agarwal, N., Adjeroh, D.: Assessing bias in YouTube’s video recommendation algorithm in a cross-lingual and cross-topical context. In: Thomson, R., Hussain, M.N., Dancy, C., Pyke, A. (eds.) Social, Cultural, and Behavioral Modeling. LNCS, vol. 12720, pp. 71–80. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-80387-2_7
12. Cano, P., Celma, O., Koppenberger, M., Buldu, J.M.: Topology of music recommendation networks. Chaos: Interdisc. J. Nonlinear Sci. **16**(1), 013107 (2006)

13. Boratto, L., Marras, M., Faralli, S., Stilo, G.: International workshop on algorithmic bias in search and recommendation (Bias 2020). In: Jose, J.M., et al. (eds.) ECIR 2020. LNCS, vol. 12036, pp. 637–640. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45442-5_84
14. Verma, S., Gao, R., Shah, C.: Facets of fairness in search and recommendation. In: Boratto, L., Faralli, S., Marras, M., Stilo, G. (eds.) Bias and Social Aspects in Search and Recommendation, vol. 1245, pp. 1–11. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-52485-2_1
15. Rosen, S.: The economics of superstars. *Am. Econ. Rev.* **71**(5), 845–858 (1981)
16. Mehrotra, R., McInerney, J., Bouchard, H., Lalmas, M., Diaz, F.: Towards a fair marketplace: counterfactual evaluation of the trade-off between relevance, fairness and satisfaction in recommendation systems. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 2243–2251 (2018)
17. Boratto, L., Fenu, G., Marras, M.: The effect of algorithmic bias on recommender systems for massive open online courses. In: Azzopardi, L., Stein, B., Fuhr, N., Mayr, P., Hauff, C., Hiemstra, D. (eds.) Advances in Information Retrieval, vol. 11437, pp. 457–472. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-15712-8_30
18. Olteanu, A., Castillo, C., Diaz, F., Kıcıman, E.: Social data: biases, methodological pitfalls, and ethical boundaries. *Front. Big Data* **2**, 13 (2019)
19. Bartley, N., Abeliuk, A., Ferrara, E., Lerman, K.: Auditing algorithmic bias on Twitter. In: 13th ACM Web Science Conference 2021, pp. 65–73 (2021)
20. Faddoul, M., Chaslot, G., Farid, H.: A longitudinal analysis of YouTube’s promotion of conspiracy videos. *arXiv preprint* <https://arxiv.org/abs/2003.03318> (2020)
21. Hussein, E., Juneja, P., Mitra, T.: Measuring misinformation in video search platforms: an audit study on YouTube. *Proc. ACM Hum.-Comput. Interact.* **4**(CSCW1), 1–27 (2020)
22. Roth, C., Mazières, A., Menezes, T.: Tubes and bubbles topological confinement of YouTube recommendations. *PloS one* **15**(4), e0231703 (2020)
23. Mirza, B.J., Keller, B.J., Ramakrishnan, N.: Studying recommendation algorithms by graph analysis. *J. Intell. Inf. Syst.* **20**(2), 131–160 (2003). <https://doi.org/10.1023/A:1021819901281>
24. Leskovec, J., Singh, A., Kleinberg, J.: Patterns of influence in a recommendation network. In: Ng, W.K., Kitsuregawa, M., Li, J., Chang, K. (eds.) Advances in Knowledge Discovery and Data Mining, vol. 3918, pp. 380–389. Springer, Heidelberg (2006). https://doi.org/10.1007/11731139_44
25. Marcoux, T., Agarwal, N., Erol, R., Obadimu, A., Hussain, M.N.: Analyzing cyber influence campaigns on YouTube using YouTubeTracker. In: Çakırtaş, M., Ozdemir, M.K. (eds.) Big Data and Social Media Analytics, pp. 101–111. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-67044-3_5
26. Kready, J., Shimray, S.A., Hussain, M.N., Agarwal, N.: YouTube data collection using parallel processing. In: 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 1119–1122. IEEE (2020)
27. Google Developers (2020). YouTube Data API, Google, <https://developers.google.com/youtube/v3>
28. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the web. Stanford InfoLab (1999)
29. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech: Theory Exp.* **2008**(10), P10008 (2008)
30. Lambiotte, R., Delvenne, J.C., Barahona, M.: Laplacian dynamics and multiscale modular structure in networks. *arXiv preprint* <https://arxiv.org/abs/0812.1770> (2008)

31. Malvestio, I., Cardillo, A., Masuda, N.: Interplay between k-core and community structure in complex networks. *Nat. Sci. Rep.* **10**(1), 1–12 (2020)
32. Erdős, P., Hajnal, A.: On chromatic number of graphs and set-systems. *Acta Math. Acad. Sci. Hung.* **17**(1–2), 61–99 (1966). <https://doi.org/10.1007/BF02020444>
33. Seidman, S.B.: Network structure and minimum degree. *Soc. Netw.* **5**(3), 269–287 (1983)
34. Clauset, A., Shalizi, C.R., Newman, M.E.: Power law distributions in empirical data. *SIAM Rev.* **51**(4), 661–703 (2009)

Diffusion and Epidemics



Modeling the Spread of COVID-19 Over Varied Contact Networks

Ryan Solorzano and Theresa Migler^(✉)

California Polytechnic State University, San Luis Obispo, CA 93410, USA
{rsolorza,tmigler}@calpoly.edu

Abstract. When attempting to mitigate the spread of an epidemic without the use of a vaccine, the implementation of an effective testing and quarantining strategy on a population has the potential to make a large impact on the spread of the disease. Yet, testing and quarantining becomes difficult when a portion of the population are asymptomatic spreaders of the disease. This research simulates the transmission of COVID-19 using five different contact networks. In these simulations, several testing and quarantining strategies are implemented with a varying number of tests available to the population per day. These strategies include a random testing strategy and several testing strategies that use underlying properties of the graph. This research found many of the strategies had a similar performance to randomly testing the population, save for testing by degree and testing the cliques of the graph which was found to consistently outperform other strategies, especially on networks that are more dense.

Keywords: COVID-19 · Disease simulation · Graph theory

1 Introduction

Mankind was not prepared for the COVID-19 pandemic, and unless we learn from it, we will not be ready for the next pandemic. There are many behavior changes we can implement as a society which have been proven to slow the spread of a disease, such as wearing masks and social distancing. Yet, while these behavioral changes lower how likely it is to transmit the disease, they do not involve the isolation of individuals we think are sick. Furthermore, the testing/quarantining plan in the United States as a whole is mostly voluntary and relies on anecdotal information of being able to list the people one was in contact with before being confirmed to have COVID-19. Duke University has showcased the effectiveness of a well implemented testing plan among a community, with their population showing much smaller numbers of infection relative to their surrounding community [2]. The efficacy of their response was credited in large part due to their “aggressive testing” strategy, when combined with their pushes for social distancing, mask mandates, hand washing, and more.

This research aims to take advantage of knowledge of the contacts of individuals in order to more intelligently test for COVID-19. This research uses five previously recorded contact networks, and develops testing intervention strategies. The testing strategies used in this research include randomly testing individuals, uniform testing, and testing based on graph theory metrics and algorithms from the contact network of individuals.

2 Background

2.1 Graphs

For this research, we will be focusing on *simple graphs*, meaning that each node can only be connected to another at most once, and there is no weight associated with these connections. The *degree* of a node is defined as the number of edges connected to it. A *clique* subgraph is one where each node has an edge to the other. A *k-core* subgraph is defined as every node having degree k or more. *Betweenness centrality* is a measure of the “centrality” of a node, meaning how many of the shortest paths go through it. *PageRank* is a very popular algorithm developed primarily by Larry Page from Google to rank the likelihood of an individual visiting a webpage.

2.2 Epidemiological Terminology

There are several terms which are used in epidemiology which need to be defined in the scope of this research, which include R_0 , *herd immunity*, *false negatives and positives*, *false positive rate*, and *false negative rate* [8,9]. R_0 is sometimes referred to as the basic reproduction number, or how many people we can expect an infectious individual to infect. The R_0 value differs across diseases, and refers to the spread of a disease *initially* [4]. As a disease spreads, this number changes and may be simply referred to as the R value of a disease. *Herd immunity* is the percentage of the population that must be immune to the disease in order for it to not persist. In terms of R_0 , if a population is at the herd immunity threshold, $R_0 < 1$ [8]. A *false negative* result means that an individual with a disease incorrectly receives a negative test result for a disease. Conversely, a *false positive* result means that an individual without a disease receives a positive result for the disease. In the study of medical diagnosis, there are two statistical features of a medical test that are important to look at: the *false positive rate* and the *false negative rate*. The false positive rate is the probability that a test result is positive given an individual is healthy, and the false negative rate is the probability that a test result is negative given that an individual is sick.

2.3 RT-PCR Based Tests

When determining if an individual has contracted a disease, a medical test must be conducted. A widely available test used for detecting SARS-CoV-2 is the

Reverse Transcriptase Polymerase Chain Reaction test, or RT-PCR [7]. This test is often administered as a swab in the nasal region, although other methods of extracting samples may include saliva, throat samples, or other bodily fluids [11]. To simulate this test, we use statistical information from the RT-PCR based test. This research attempts to approximate the false negative rates found by Kucirka et al. by using the values found in Table 1. It should be noted that they found a considerable uncertainty in their numbers, however for the sake of this experiment we will assume the numbers found are correct. It should be noted that Kucirka et al. did not try to determine the false positive rate of the tests, however Dr. Shmerling of Harvard Medical Health says that we may expect this value to be at or near zero, as any false positives are likely due to lab equipment error [11]. So, for this simulation the false positive rate will be a constant 2%. Finally, for these simulations we assume immediate feedback from the test results, which more closely resembles antigen tests rather than RT-PCR [11]. However, this assumption is made because the simulations were designed to showcase the effectiveness of the testing strategies rather than the tests themselves. Table 1 summarizes the false negative rates used in this simulation.

Table 1. The False Negative rates of RT-PCR based testing, where Day is days since exposure, and FNR is the False Negative Rate, or percent chance someone will get a false negative result

Day	1	2	3	4	5	6	7	8	9	10
FNR	100	100	97.7	71.0	38.7	24.8	20.1	19.1	20	22
Day	11	12	13	14	15	16	17	18	19	20
FNR	25	28.6	32.5	36.8	41.2	45.5	49.6	53.5	57.0	60.2

3 Related Works

A work by St-Onge et al. first tried to model a more realistic spread of a hypothetical disease using an improved SIR model [13]. Given the timing of the experiment, it may be assumed that this model was loosely based on COVID-19. The authors then showcased the importance of intervention strategies in mitigating the spread of disease in networks with higher-order structure (i.e. a large community). The evaluation of the intervention strategies is of particular interest in this research since this paper also explores intervention strategies to mitigate the spread of a disease. Imai et al. was one of the first studies to show that the transmission of SARS-CoV-2 was self-sustaining, or that its R value is greater than 1 [4]. Additionally, they calculated the R_0 value of COVID-19 to be 2.6. This study was conducted in Wuhan, China at the beginning of the COVID-19 pandemic, January 2020 and observed all the estimated amount of cases prior to the publication of the paper. This paper is of particular interest because it gives us the realistic values of the spread of COVID-19 in the absence of disease intervention

strategies such as mask wearing which to compare the results of the simulations of this research to. Kucharski et al. explored the effect of several intervention strategies on the effective R_0 value of COVID-19 [6]. This work found that a combined testing and tracing strategy was the most effective at lowering the R value of a disease. Similar to Imai et al., the evaluation of these intervention strategies is important with this research as this study explores similar intervention strategies to this research. Siu et al. attempt to mitigate the spread of a disease by creating vaccine intervention strategies using graph theory [12]. Additionally, rather than simulating the epidemic using a mathematical model, they also use a contact network gathered from Copenhagen [14]. This work also implements vaccination strategies using underlying reasoning which is similar to a testing/quarantining strategy. G'enois et al. explore if using co-location information can be down sampled to accurately model a real life contact network [3]. Here, co-location data is defined as two individuals being in the same general area such as a room. This paper used the SocioPatterns datasets and was able to show that there was no down-sampling technique which is able to accurately model real world interactions across any scenarios. This is helpful in the realm of this research, because there are many more contact networks that look more like a co-location interaction rather than a face to face interaction, so this paper shows that using these networks is not as accurate. Estrada et al. have a very in-depth paper on how to mathematically model SARS-CoV-2 using a modified SIR model [1]. While I'm more focused on modeling using contact networks rather than mathematical models, this paper is still extremely helpful in creating my modified compartmental model, and for parameters for my simulation such as probability of infection. Shah et al. explore the correlation between the super spreaders of a disease and the PageRank of those spreaders [10]. One of the results of this study found that the nodes with a high PageRank value contain a higher proportion of super spreaders than the nodes with lower PageRank values. It should be noted that this study was done in the context of the spread of disease among herds of livestock, however the results should still apply in a human contact network.

4 Network Descriptions

The experiments done in this research use contact networks gathered by SocioPatterns. SocioPatterns is a collaboration between researchers and developers across the world to collect various contact network data for networks ranging from Baboons' interactions to interactions among people within a hospital [3]. All of the data is collected using the same system where each participant wears an RFID tag and reader which records if two individuals come into contact. In the case of this network, a contact is defined as the readers of both individuals register the RFID tag of the other for a 20 s time window. This collaboration states that two individuals must be within a 1.5 m of each other to record a contact, or about 5 ft. However, intensity is not specified meaning we must therefore assume that all contacts are equal and is sufficient duration and distance to

Table 2. A description of each contact network used for this research [3].

Network	Year	Participants	Duration
Workplace 1	2013	92	2 weeks
Workplace 2	2015	232	2 weeks
High school	2013	326	1 week
Conference	2009	403	2 days
Preschool	2009	242	2 days

Table 3. Properties of each contact network. Values are averaged over all the days of the study [3].

Network	Average degree	Network density	Clique number
Workplace 1	2.9	0.030	4.4
Workplace 2	6.4	0.028	7.6
High school	13.5	0.041	9.4
Conference	28.8	0.072	11.0
Preschool	47.3	0.196	22.5

transmit COVID-19 from an infectious individual to a susceptible one. Table 2 summarizes each contact network used in this research and Table 3 summaries the properties of each network.

5 Simulation

The bucket model for this research is the SEQIR model shown in Fig. 1. The model starts with the entire network in the SUSCEPTIBLE bucket, except one randomly chosen person is moved to the EXPOSED bucket. Once a person is in the EXPOSED bucket, they are given a certain amount of time before they are moved to INFECTIOUS. From the INFECTIOUS bucket, if a person has a recorded interaction with someone in SUSCEPTIBLE, they have a 3% chance of infecting the person and moving them to EXPOSED. This value falls within the secondary contact range, used by Kucharski et al. [6], and was chosen based on the high school network having a R_0 value of 2.69 without any interventions, which is similar to that found by Imai et al. [4]. Then, after being infectious for a varying amount of time, a person is moved to the REMOVED bucket, where they are unable to move out. It should be noted that there has been reports of multiple COVID-19 infections in a single patient [5], however this simulation assumes one infection is sufficient due to the rare nature of a second reinfection in the time span of the simulations.

Once a day, tests are performed on the eligible population (those in the SUSCEPTIBLE, INFECTIOUS, and REMOVED buckets), the number of which depends on the simulation. The probability for a SUSCEPTIBLE individual to test positive is a constant 0.02%, and the probability for an EXPOSED and INFECTIOUS individual to test positive is based on the RT-PCR statistics in Subsect. 2.3. If a positive test is given, the individuals are moved to the QUARANTINED bucket for two weeks, where we assume a perfect quarantine. Then, if the individual was sick, they are moved to REMOVED, or if they got a false positive they are moved to SUSCEPTIBLE.

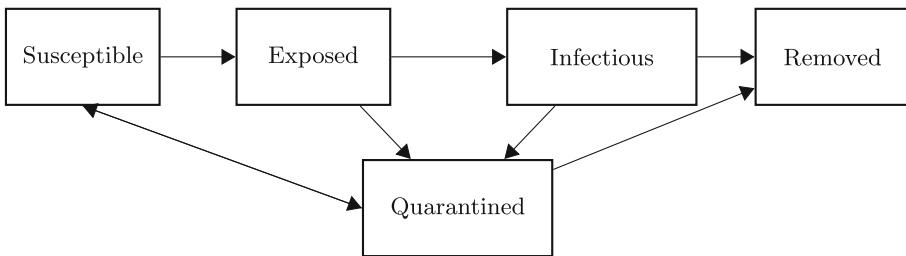


Fig. 1. Proposed SEQIR Model

6 Intervention Strategies

This research explores seven different testing strategies with a varying amount of available tests. In the simulation, the testing strategy is implemented once a day in the morning. Additionally, only the eligible population is tested, meaning only the population in the SUSCEPTIBLE, EXPOSED, and INFECTIOUS buckets.

Another important note is that these strategies do not differentiate between asymptomatic and symptomatic infectious individuals. So, any infectious individual has the same probability of testing positive given that the day they were infected was the same. Additionally, if a person is symptomatic, they will still not be tested unless the following strategies dictate that it is necessary.

No Testing. In order to create a control which to compare the other testing strategies, simulations without any intervention were run. These simulations omitted any testing, and therefore no one was ever moved to the quarantined bucket.

Random Batch Testing. This strategy is the simplest which one can come up with which tests a random subset of the population up to the number of tests available. While rudimentary, this actually somewhat simulates the initial testing in the United States. Anecdotally, there were only a limited number of

tests available in the early stages of the COVID-19 pandemic, and there was not a strong system in place for asymptomatic individuals to be tested. Therefore, we could view this testing strategy as an approximation of the early development of testing in the US. It can also give us a baseline to compare other intervention strategies to; if a strategy does worse than this one, then we can safely say that it is not an effective strategy.

Individual Testing. For the sake of this simulation with a varying amount of tests, we will say that people who have not been tested the longest will be given the tests first.

Testing Using Graph Properties and Algorithms. For each of these strategies, a timeslice of the previous day is recorded, and a graph is created.

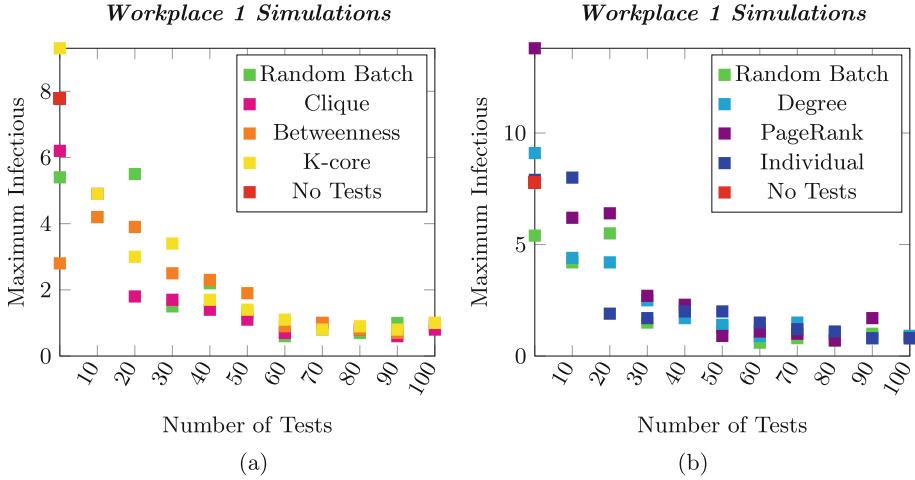
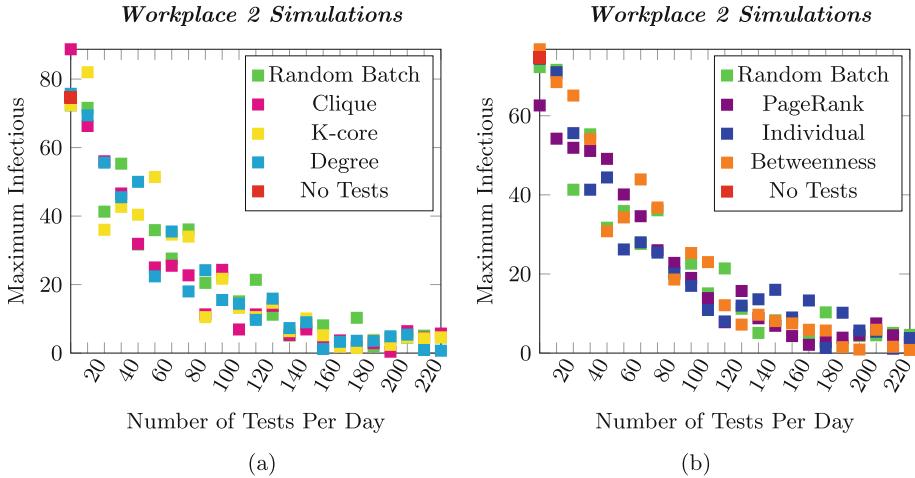
Degree: This strategy uses the timeslice graph, and tests the people in order from highest degree. *Clique* and *k-Core*: These strategies uses the timeslice graph, and tests people in order from highest clique number or k-core value. *Betweenness Centrality:* This strategy uses the timeslice graph, and tests people in order from high centrality score.

PageRank: This strategy uses the timeslice graph, and tests people in order from highest PageRank score. For this experiment, we are reinterpreting the algorithm from people moving across websites, to diseases moving from person to person.

7 Results

Here, we consolidate the results from each simulation. For each contact network, all seven testing strategies were simulated with the number of tests conducted per day varying from zero to the size of the population. Furthermore, each test size experiment was run ten times, and the results were recorded and consolidated in this section. The results of the experiments are given in the format of the maximum number infected.

Figure 2 through Fig. 6 take the peak number of people infected at a given time and averages the value across all ten simulations for a given contact network and testing strategy. This gives a single data point for a given testing strategy for the number of tests given. This is done for all simulations and are plotted with number of tests on the X axis and average peak number infectious on the Y. However, since there are so many testing strategies, these plots are split up into two plots with the three best testing strategies and random batch on the left, and the three worst and random batch on the right Figs. 3 and 4.

**Fig. 2.** Results for each strategy on the Workplace 1 contact network**Fig. 3.** Results for each strategy on the Workplace 2 contact network

The method by which the “best” strategies was determined was a simple scoring system. For each point in the graph, the point with the smallest value for maximum infectious was given a score of 6, the second smallest a score of 5, and so on. Table 4 summarizes these scores.

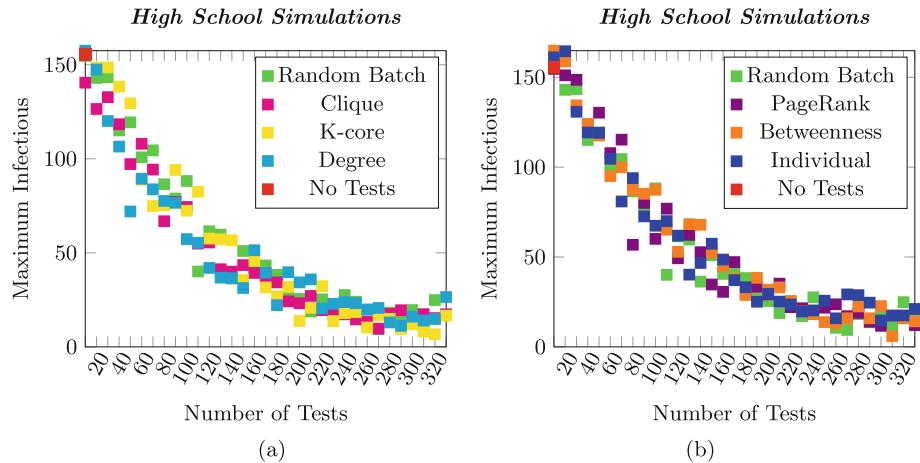


Fig. 4. Results for each strategy on the High School contact network

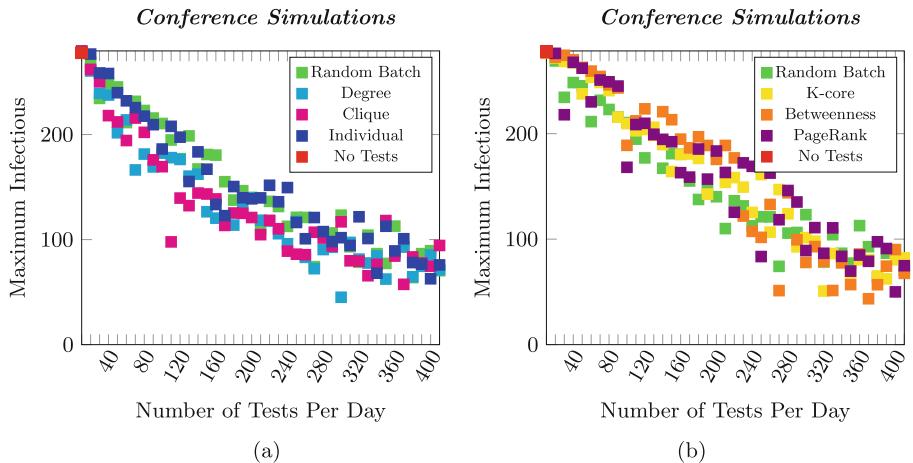


Fig. 5. Results for each strategy on the Conference contact network

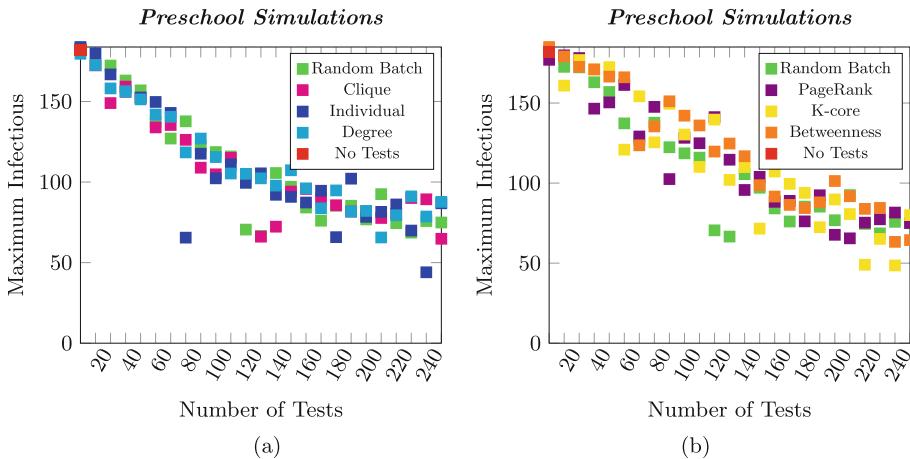


Fig. 6. Results for each strategy on the Preschool contact network

8 Evaluation

8.1 Testing Using Graph Properties

Degree. This is the first strategy we see consistently outperform the random batch strategy. In fact, based on Table 4, we see that this strategy is consistently one of the best strategies of this research. Furthermore, we can see that as the average degree and density increases, this strategy only does better. This fact is especially showcased in Fig. 5. The success of this strategy likely is due to the repetitive nature of these experiments, with the replays consistently giving people similar degree values. However, this fact mirrors real life as people tend to follow a consistent routine and are consistently exposed to a similar number of people day to day.

Table 4. A summary of the score for each strategy across each contact network when considering the maximum number infected

	Workplace 1	Workplace 2	High school	Conference	Preschool
Random	57	87	131	154	112
Individual	38	91	110	153	108
Degree	42	96	146	225	102
Clique	65	104	158	223	115
K-Core	42	99	152	144	96
Betweenness	49	80	112	136	70
PageRank	38	94	120	114	100

Clique. Another exciting result, as we see testing by clique number not only matches the performance seen by degree, but surpass it in the higher density networks. Using the results of Table 4, we see that this testing strategy appears to be the most effective across the table.

K-Core. While it appears that this strategy performed well, we can see that as the average degree and density of the graph increased, the performance of this strategy decreased. In fact, in the Conference and Preschool networks, this strategy performed worse than random batch testing.

Betweenness Centrality. A slightly different and less intuitive approach, this strategy attempted to slow the spread by targeting those individuals who were “in between” portions of a graph. Ideally, these would be the people who connect different portions of the contact network, i.e. those who carry the disease across social groups.

PageRank. Another unique approach, the motivation of this strategy was to try to predict where the disease may be. However, this is another strategy that was typically outperformed by random batch testing.

9 Conclusion

This research explored numerous testing intervention strategies, and determined the viability of each strategy across various contact networks. The complexity of each strategy ranged from simply randomly testing a portion of the population every day to adapting an algorithm originally intending for ranking the relevance of web pages.

Even with the success of the testing strategies, we must also reflect upon the shortcomings of the simulations. First was the assumption that the behavior of individuals would not change with an outbreak of an extremely infectious disease, save for quarantining if an individual tests positive. While this behavior to reduce the spread such as mask wearing and social distancing may be able to be adapted to these contact networks, it was intentionally omitted as this research’s primary focus was to evaluate the performance of the testing strategies, not to evaluate the effectiveness of human behavior. Leaving out symptomatic and asymptomatic infectious individuals was done for the same reason; including symptomatic individuals would likely improve the testing and quarantining process as the strategies would be forced to always test symptomatic people.

10 Future Work

Given the results of the experiments, we must remember the assumptions made with this experiment. Much of the potential future work may change these

assumptions in order to make a more realistic simulation and explore a different aspect of disease transmission. The first assumption is that the behavior of the contact networks would remain the same in a pandemic. Additionally, we assume a constant infection probability, when in reality it differs depending on the individual infected, the distance of interactions, if one or both of the individuals are wearing a face covering, and so on. One of the most notable assumptions is that there is no distinguishment between the asymptomatic infectious individuals and symptomatic infectious individuals. In the simulations for this research, every individual who is infected is treated as asymptomatic in that they do not self-quarantine or get tested if they feel sick. Lastly, we assumed a closed network for this research, meaning there is no outside influence for infections. However, an improvement that may be made is to combine the networks and make households to make the final simulation more granular. All of these considerations allow interesting directions to be taken for future work.

Finally, since the beginning of this research, a vaccine has been developed. Experiments that simulate vaccinating the population with the 2-dose vaccine like Moderna or Pfizer with varying efficacy would be an interesting direction to take this research as well. In short, this research could go in many directions and is limited to what aspect of the simulation is wanted to make more realistic or how elaborate the proposed intervention strategy would be.

References

1. Estrada, E.: COVID-19 and SARS-CoV-2. Modeling the present, looking at the future. *Phys. Rep.* **869**, 1–51 (2020). <https://www.sciencedirect.com/science/article/pii/S0370157320302544>
2. Ferreri, E.: Students ‘Nothing Short of Phenomenal’ in Minimizing COVID Spread, November 2020. <https://today.duke.edu/2020/11/students-nothing-short-phenomenal-minimizing-covid-spread>
3. Génois, M., Barrat, A.: Can co-location be used as a proxy for face-to-face contacts? *EPJ Data Sci.* **7**(1), 11 (2018). <https://doi.org/10.1140/epjds/s13688-018-0140-1>
4. Imai, N., Cori, A., Dorigatti, I., Baguelin, M., Donnelly, C.A., Riley, S.: Report 3: transmissibility of 2019-nCoV, p. 5, January 2020
5. Iwasaki, A.: What reinfections mean for COVID-19. *Lancet Infect. Dis.* **21**(1), 3–5 (2021). [https://www.thelancet.com/journals/laninf/article/PIIS1473-3099\(20\)30783-0/abstract](https://www.thelancet.com/journals/laninf/article/PIIS1473-3099(20)30783-0/abstract)
6. Kucharski, A.J., et al.: Effectiveness of isolation, testing, contact tracing, and physical distancing on reducing transmission of SARS-CoV-2 in different settings: a mathematical modelling study. *Lancet Infect. Dis.* **20**(10), 1151–1160 (2020). <https://linkinghub.elsevier.com/retrieve/pii/S1473309920304576>
7. Kucirka, L.M., Lauer, S.A., Laeyendecker, O., Boon, D., Lessler, J.: Variation in false-negative rate of reverse transcriptase polymerase chain reaction-based SARS-CoV-2 tests by time since exposure. *Ann. Intern. Med.* (2020). <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7240870/>
8. Mullner, R.M.: Epidemiology. <https://www.britannica.com/science/epidemiology>
9. Rakel, R.E.: Diagnosis - Mental examination, November 2018. <https://www.britannica.com/science/diagnosis>

10. Shah, N., Shah, H., Malensek, M., Pallickara, S.L., Pallickara, S.: Network analysis for identifying and characterizing disease outbreak influence from voluminous epidemiology data. In: 2016 IEEE International Conference on Big Data (Big Data), pp. 1222–1231, December 2016
11. Shmerling, R.H.: Which test is best for COVID-19? August 2020. <https://www.health.harvard.edu/blog/which-test-is-best-for-covid-19-2020081020734>
12. Siu, C., Migler, T.: Vaccination strategies on a robust contact network. In: Cherifi, H., Gaito, S., Mendes, J.F., Moro, E., Rocha, L.M. (eds.) Complex Networks. SCI, vol. 881, pp. 313–324. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-36687-2_26
13. St-Onge, G., Thibeault, V., Allard, A., Dubé, L.J., Hébert-Dufresne, L.: Social confinement and mesoscopic localization of epidemics on networks. Phys. Rev. Lett. **126**(9), 098301 (2021). [arXiv: 2003.05924](https://arxiv.org/abs/2003.05924)
14. Stopczynski, A., et al.: Measuring large-scale social networks with high resolution. PLoS ONE **9**(4), e95978 (2014). <https://doi.org/10.1371/journal.pone.0095978>



Benchmarking Optimal Control for Network Dynamic Systems with Plausible Epidemic Models

Yaroslav V. Salii^{1,2(✉)}

¹ McGill University, Montreal H3A 0E9, Canada
yvsalii@cim.mcgill.ca

² Krasovskii Institute of Mathematics and Mechanics UB RAS, ul. S. Kovalevskoi 16, 620108 Yekaterinburg, Russia

Abstract. The sheer dimension of *network dynamic systems* adds a challenge of scale to synthesizing *optimal control*, which the techniques such as mean-field approximation, reinforcement learning, and graphon mean field games attempt to overcome. We propose to use *compartmental metapopulation epidemic models* derived from open data to benchmark these advanced approaches on an important problem with intuitive visualization options such as choropleth maps. To this end, we formalize a procedure for generating plausible instances of such models with 1–64,735 nodes based on open census data for the contiguous U.S., each with a network of *daily commute* and *airplane travel*, coupled with a formal *aggregation* routine enabling a view of the same geography at different resolutions, illustrated by merging the 2,072 census tracts in Oregon and Washington states, together with their travel networks, into 75 *county-level* nodes, 23 “airport service area” nodes, and 2 nodes for *states* themselves. These four cases, and ten other, are then put through 180-day “patient zero” scenarios in a Metapopulation SIR Model with per-node “lockdown level” control, with the objective of minimizing the cumulative number of infections and the *lockdown level*. The optimal control is derived through the Pontryagin Maximum Principle and numerically computed with the *forward-backward sweep* method. To ensure reproducibility, the instance generator, solver, and visualization routines are available at <https://github.com/yvs314/epi-net-m>.

Keywords: Optimal control · Very large-scale networks · Social distancing · Metapopulation epidemic model

Introduction

There are many options to tackle the curse of dimensionality in network dynamic systems e.g. by exploiting the network structure, be it a sparse representation of

This work was supported by AFOSR grant FA9550-19-1-0138 and ARL grant W911NF1910110.

the graph or node degree-aware mean-field approximation such as [14, Chs. 4, 5]. An intriguing option is recourse to *graphons* [17], continuous limit objects of graph sequences. The options for using those in control for linear systems are discussed in [12], and applications to epidemics are proposed in [3, 9].

One would like to *demonstrate* that such approaches can save computation effort in practical problems, to put them through a series of problem instances with increasing number of nodes derived from some realistic data, and compare this to a “baseline” method, which would work with an explicit network representation. This can be done with *metapopulation epidemic models*, which treat the propagation of infection over a transportation network through several nodes, see the review in [20] and results on asymptotics in [19].

The data for network epidemic models, whether compartmental or agent-based, is usually sourced from census and statistics offices, see e.g. [4, 8]. Another powerful approach is to derive the nodes’ interconnections from the time series data of an actual epidemic [1]. The objective of these works is typically forecasting, which requires extensive parameter fitting and often precludes a comparison of different solution methods, with notable exceptions in [2, 5], which pit agent-based models against the “aggregated” metapopulation models.

The importance of network structure in ODE-based models is discussed in e.g. [23, 27]. In [18], several network epidemic models, their linearizations, and approaches to compute “spatially heterogenic” *optimal lockdown* control, together with the influence of different network structures on the control, are discussed, and the results are applied to model the lockdown and infection rates in the 62 *counties* in the New York state over several hundred days. In [7], a “weekly-constant” model predictive control scheme is derived and applied to model the mitigation of a COVID-19 outbreak for the 28 regions of Italy by non-pharmaceutical intervention. The largest node number in optimal control statement the author is aware of is the 298-node instance in [16].

Among non-network statements, let us note [22], which accounts for the development of health infrastructure, quarantine measures, mobility restrictions, and social distancing, as well as the population’s increasing fatigue with these measures, implemented as exponential discounting in the objective function.

Contribution

- (i) We formalize the procedure of *generating* the “instances” with the given number of nodes between 1 and 64,735, each endowed with appropriate short-range and air travel networks and per-node populations, derived from open, publicly available data [6, 8] and the procedure of *aggregating* such data, which enables a view of the same geography at different resolutions.
- (ii) We propose a rudimentary optimal control scheme for metapopulation network SIR model, derived from that of [22] by (a) adding the network structure and per-node populations; (b) reducing the compartments to just SIR; (c) considering only the “securing social interactions” control.

- (iii) We test the performance of *forward-backward sweep* [15, Ch. 4] with variable-step ODE solver as a numerical solution method for (ii) on 14 instances from 2 to 4,830 nodes with varying levels of aggregation on 180-day “patient zero” scenarios.

Paper Structure In Sect. 1, we formalize the generation of instances varying in scale and resolution, with a case study on Northwestern U.S. states in Sect. 1.4. Section 2 presents the metapopulation network SIR model and Sect. 3 describes the optimal control scheme for the latter; finally, Sect. 3.3 sets forth the computation results on 180-day patient zero scenarios.

1 Generating Model Instances of Varying Scale and Resolution

We start by conceptually separating the *model*—the compartments, time scale, and parameters—and the *data set*, a set of elementary population points (nodes) with geographical coordinates and the data on mobility between them. Then, an *instance* is a partition of a subset of the data set: some nodes may be omitted, some may be merged; the latter are treated as a single node where homogeneous mixing is assumed; the mobility data are aggregated. Finally, a *scenario* is an instance coupled with a *model* and *initial values*.

1.1 The Data Set

Like in [4], we consider three aspects of the data: population, short-range travel, and long-range travel; in addition, we consider varying levels of aggregation.

Our atomic **populated areas** are the *census tracts* in the contiguous U.S.—“small, relatively permanent statistical subdivisions of a county”—as considered in the 2010 Census, for each of which its *geographical coordinates* (latitude and longitude) and *population* are available. We take these data from the FluTE model [8], see <https://github.com/dlchao/FluTE>. There are 64,735 census tracts in this database, with the median population 4,024, minimum at 1 person, and maximum at 36,146.

The **short-range travel** is the number of people who commute between them each day. These data we also take from the FluTE repository; the data is based on *worker flow* between census tracts, originally obtained from the U.S. Census Transportation Planning Package, from which the commutes over 160.93 km (100 mi) were eliminated [8].

The **long-range travel** is how many people arrive and depart by air each day for each pair of nodes, however, such data is not readily available on tract-to-tract level. As a substitute, we take the total numbers of passengers on domestic flights in the U.S., published by Bureau of Transportation Statistics (BTS) [6]. We use the data for the year 2019 and divide it by 365 to get average passengers-per-day between each airport (AP).

This data set, coupled with an aggregation procedure of Sect. 1.3, enables us to consider instances on 1–64,735 nodes at various *geographical scopes*, which

should let the user find the “saturation point” of a solution method, i.e. the maximum number of nodes at which it is still practicable to run it on a given computer. The range of model sizes is hopefully sufficient to get sensible solutions with the methods the error of which decreases with model size.

1.2 Distributing Air Traffic to Population Units

The U.S. Bureau of Transportation Statistics (BTS) publishes a data set [6] that is essentially a list of triples *departure AP–arrival AP–no. passengers*. We assume the travelers from a given census tract always fly through one *designated AP*, i.e., the inter-AP passenger flows are distributed between the census tracts in each AP’s “service area,” which is similar to the GLEAM approach [4].

Denote the set of all *nodes* by ND . For two APs a, b , let $\text{psg}_{a,b}$ denote the average number of *passengers per day* between them. By abuse of notation, for two *nodes* $a, b \in \text{ND}$, we reuse $\text{psg}_{a,b}$. For each node $a \in \text{ND}$, assume everyone traveling to or from a by air to pass through its *designated AP*, denoted dsg_a . Assume the fraction of passengers traveling through dsg_a is proportional to the node’s *population* pop_a , and call this proportion node a ’s *airport share* in dsg_a ,

$$\text{shr}_a \triangleq \frac{\text{pop}_a}{\sum\{\text{pop}_c \mid c \in \text{ND} \wedge \text{dsg}_c = \text{dsg}_a\}}.$$

We assume there is no air travel between two nodes with the same *designated AP*. For two nodes $a, b \in \text{ND}$ with distinct *designated APs*, the passenger flow is obtained by weighing the number of daily passengers from dsg_a to the dsg_b by the *airport shares* of the nodes: $\text{psg}_{a,b} \triangleq \text{psg}_{\text{dsg}_a, \text{dsg}_b} \cdot \text{shr}_a \cdot \text{shr}_b$.

1.3 Aggregating the Nodes and Their Connections

A hierarchy of aggregation from census tracts to counties to states lets one compare different “resolutions” of models on the same geography with the same total population, e.g. to get an explicit count of by how much does assuming *homogeneous mixing* in a geographically large and populous areas *overestimates* the effective infection rates, which would be tempered by accounting for connection intensities in a more granular model; see a discussion of this effect in [23, Sec. 8].

It is straightforward to aggregate census tracts by administrative subdivision: each of the 64,735 census tracts is identified by a numeric code assigning it to a specific U.S. *county* and *state*, e.g. census tract 41.001.950100 is in state 41, Oregon, and county 001, Baker Cty. The whole data set is thus partitioned into 3,109 *counties* or 49 larger entities (48 states and 1 federal district). Another option is the approach of [4]: aggregate all nodes served by a single AP.

Let $\widetilde{\text{ND}}$ be a *partition* of the node set ND . Denote the daily passengers between $\tilde{a}, \tilde{b} \in \widetilde{\text{ND}}$ by $\widetilde{\text{psg}}_{\tilde{a}, \tilde{b}}$, and set it to the total number of passengers between the *elements* of \tilde{a} and \tilde{b} : $\widetilde{\text{psg}}_{\tilde{a}, \tilde{b}} \triangleq \sum_{a \in \tilde{a}, b \in \tilde{b}} \text{psg}_{a,b}$. The aggregated number of *daily commuters* $\widetilde{\text{cmt}}_{\tilde{a}, \tilde{b}}$ is computed similarly, except the underlying $\text{cmt}_{a,b}$ is set to 0 whenever a and b belong to the same element of the partition.

1.4 Instances from Census Tracts in Oregon and Washington States

We describe the data processing with a case study on the 2,072 *census tracts* that have either “41” (Ore.) or “53” (Wash.) as their state identifier; assume these make up the node set ND . These are illustrated on Fig. 1.

According to the BTS data [6], there are at least some degree of passenger service in 1,248 APs in 2019. Of these we retain the 543 that had at least 2,500 total (incoming plus outgoing) boardings in 2019, similar to the FAA qualification of “Commercial Service” APs. Their *geographical coordinates* we get from Openflights Airport Database [21], matching on the APs’ three-letter IATA codes. Assume these 543 APs make up the set AP .

With geographical coordinates in hand, we assign to each node $a \in ND$ the nearest AP in terms of Earth *big circle distance*. We then compute the AP shares and the daily passengers between all of the nodes as described in Sect. 1.2; recall that the nodes with the same *designated AP* are assumed to have no air connections. There are 23 such APs, and the maximum distance from a census tract to its designated AP is 178.3 km, with the median at 19.5 km. The numbers of daily commuters between census tracts are available from the FluTE [8] project repository. Finally, we sum the daily air passengers and daily commuters to produce the *daily travelers* matrix $\hat{\mathbf{A}} \triangleq [\hat{a}_{ij} = psg_{i,j} + cmt_{i,j}]$.

We produced 16 instances with 2–9,110 nodes, which are available from the GitHub repository at <https://github.com/yvs314/epi-net-m>; the data processing was done with Julia 1.6 through `DataFrames.jl`. Table 1 describes the instances and their travel networks. For commute and air travel, the columns ρ list the percentage of *nonzero* connections; their *median* element was counted ignoring the zero elements. The `NWtra2072`, `NWcty75`, and `NWste2` are aggregated by, respectively, tract, county, and state, while in `NWap23`, all nodes served by the same *designated AP* are merged. The “administrative” instances show a steady increase in the density of the commute and travel matrices. It should be noted that while commute is more sparse at all levels of aggregation, it carries many more persons per day and thus leads to tighter coupling than the air travel. Another thing to note is that at the *by-state* aggregation (`NWste2`), the estimated air travel is nearly symmetric at about 3,400 people per day, whereas the commute is skewed, with 56,683 commuting cross-border from Washington to Oregon and only 16,139 vice versa.

Table 1. Population, air travel, and commute under four types of aggregation

Instance	Node Type	n	Population			cmt $_{a,b}$		psg $_{a,b}$	
			Min	Median	Max	ρ	Median	ρ	Median
<code>NWtra2072</code>	Tract	2,072	36	4,317	16,747	3%	10	39%	0.001
<code>NWcty75</code>	County	75	1,547	44,479	1,737,034	15%	27	43%	0.155
<code>NWap23</code>	AP service area	23	12,902	228,336	2,197,954	28%	168	15%	14.431
<code>NWste2</code>	State	2	3,421,399	–	5,894,121	100%	–	100%	–

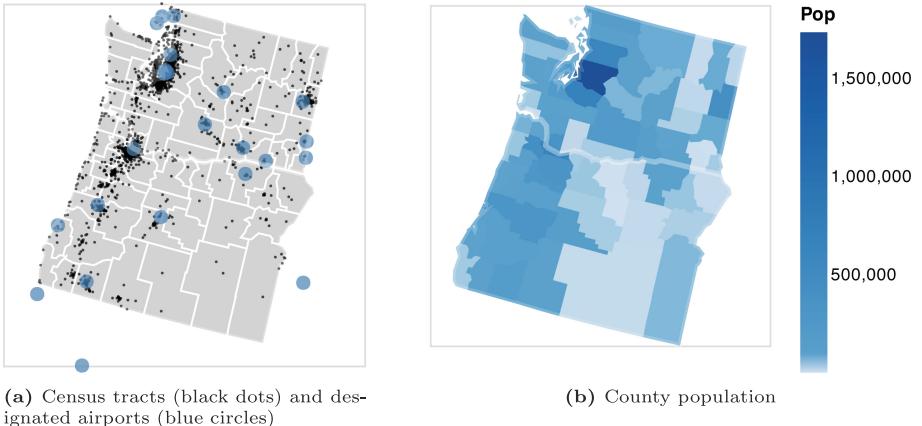


Fig. 1. Data set for Oregon and Washington states

2 Metapopulation Network SIR Model

Vector Algebra Notation

The symbol \triangleq denotes equality *by definition*. The *unity matrix* of dimension n is \mathbf{I}_n ; the n -vector with 1 at each its component is $\mathbf{1}_n$; and the n -component zero vector is denoted $\mathbf{0}_n$. Vectors are *column* by default. The *transpose* is denoted by \top , e.g. $x = [x_1, \dots, x_n]^\top$ is a *column* and its transpose $x^\top = [x_1, \dots, x_n]$ is a *row* vector. The *scalar product* of vectors x and y is denoted by *juxtaposition* $x^\top y$. The *Hadamard product* of vectors x and y is $x \odot y \triangleq [x_1 y_1, \dots, x_n y_n]^\top$. For an $[n \times n]$ matrix $\mathbf{A} = [a_{ij}]$, denote its i th row a_{i*} and its j th column a_{*j} .

2.1 Network SIR Model

We consider an n -node model on the time interval $[0, T]$, short enough that we may ignore the vital dynamics and let the node populations vector $N = [N_1, \dots, N_n]^\top$ be constant. The nodes interact through an $[n \times n]$ dimensionless *connection matrix* $\mathbf{A} = [a_{ij}]$, where a_{ij} is the *connection intensity* from node i to node j ; we do *not* assume symmetry $a_{ij} = a_{ji}$, although this is common, see e.g. [24]. To get a proper “continuation” of the single-population model, we assume $a_{ii} = 1$, which is a trivial “connection” of a node with itself. The off-diagonal elements are assumed to be *less* than 1.

A connection matrix \mathbf{A} is obtained from the *absolute* travelers-per-day matrix $\hat{\mathbf{A}} \triangleq [\hat{a}_{ij} = \text{psg}_{i,j} + \text{cmt}_{i,j}]$ from Sect. 1.4 by multiplying each its row through the *inverse* of the corresponding population, $\mathbf{A} \triangleq \text{diag}(N^{-1})\hat{\mathbf{A}}$, a procedure similar to the operation of the *transport operator* in [24, Eq. (8)].

Each person can be susceptible, infectious, or recovered (exclusively). The susceptible become infected at the *contagion rate* $\beta \in \mathbb{R}^+$, and infected recover at the *recovery rate* $\gamma \in \mathbb{R}^+$. The state of the metapopulation system is described

by three n -vectors, $s(t)$, $z(t)$, and $r(t)$, the i th components of which are node i 's fractions of, respectively, susceptible, infected, and recovered (immune) populations. In SIR model, the recovered vector r is effectively computed by closure, $r(t) \triangleq \mathbf{1}_n - s(t) - z(t)$, and may be omitted from the dynamics.

The network SIR model, in per-node and vector forms, is as follows:

$$\begin{cases} \dot{s}_i = -\beta s_i a_{*i}^\top z; \\ \dot{z}_i = \beta s_i a_{*i}^\top z - \gamma z_i; \end{cases} \quad \begin{cases} \dot{s} = -\beta s \odot \mathbf{A}^\top z; \\ \dot{z} = \beta s \odot \mathbf{A}^\top z - \gamma z; \end{cases} \quad (1)$$

the transpose $^\top$ is affixed to preserve the natural interpretation of \mathbf{A} 's coefficients, i.e. that a_{ij} is the intensity of connection from i to j . For analysis of these dynamics, see [19, Th. 5.2]. For derivation from probabilistic arguments, and the proof that this model and its discretization are *well-defined*, see [23, Secs. 3.2,4.2].

3 Network SIR with “Social Distancing” Control

We view the effect of each node's *social distancing* control $u_i(t)$ as a reduction in the node's *contagion rate* β , which gives the following optimal control problem:

$$\begin{cases} \dot{s} = -\beta(\mathbf{1}_n - u) \odot s \odot \mathbf{A}^\top z; \\ \dot{z} = \beta(\mathbf{1}_n - u) \odot s \odot \mathbf{A}^\top z - \gamma z; \end{cases} \quad u(t) \in [0, 1]^n; \quad (2)$$

with initial conditions $s(0) = s_0; z(0) = z_0$;

$$\text{minimize } J[u, x, t] = e^{r_1 T} k z^\top(T) N + \int_0^T \left(e^{r_1 t} c z(t) + e^{r_2 t} \frac{q}{2} u(t) \odot u(t) \right)^\top N dt, \quad (3)$$

where $x \triangleq [s_1, \dots, s_n, z_1, \dots, z_n]^\top$ is the combined state vector.

This system's control-related parameters are the infection's *terminal cost* $k \in \mathbb{R}^+$, the infection's *running cost* $c \in \mathbb{R}^+$, the control's *running cost* $q \in \mathbb{R}^+$, and the *fatigue rates* $r_1, r_2 \in \mathbb{R}^+$, where r_1 is associated with infections and r_2 with the toll of having to adhere to social distancing measures.

The principal difference in the objective function (3) vs. the single-population form [22] and the *node-based* SIR [16] is the fact that both *terminal* and *running* costs are *weighed by node population*, as expressed in the scalar product through the population vector N . For reasons of space, we omit the proof of *existence* of optimal control for (2)–(3), however, let us note that it follows from an application of [10, Th. 4.1].

3.1 Minimizing $J[\cdot]$ Through Pontryagin Maximum Principle (PMP)

Let $\lambda_s \triangleq [\lambda_{s_1}, \dots, \lambda_{s_n}]^\top$ and $\lambda_z \triangleq [\lambda_{z_1}, \dots, \lambda_{z_n}]^\top$ be the *costate* (adjoint) vectors for (2), with the overall costate λ as their vertical concatenation. Then, the *Hamiltonian* $H[\cdot]$ of system (2),(3) is

$$H[u, x, t, \lambda] = \left(e^{r_1 t} c z(t) + e^{r_2 t} \frac{q}{2} u^2(t) \right)^\top N - \lambda_s^\top \beta (\mathbf{1}_n - u) \odot s \odot \mathbf{A}^\top z + \lambda_z^\top [\beta (\mathbf{1}_n - u) \odot s \odot \mathbf{A}^\top z - \gamma z(t)].$$

The above is for the “minimize $H[\cdot]$, minimize $J[\cdot]$ ” form of the Pontryagin Maximum Principle (PMP), see e.g. [13, Sec. 5.1], from which we obtain the costate equations $\dot{\lambda}_s = -H_s$ and $\dot{\lambda}_z = -H_z$:

$$\begin{cases} \dot{\lambda}_s = \beta(\lambda_s - \lambda_z) \odot (\mathbf{1}_n - u) \odot \mathbf{A}^\top z; \\ \dot{\lambda}_z = -e^{r_1 t} c N + \beta \mathbf{A}(\lambda_s - \lambda_z) \odot (\mathbf{1}_n - u) \odot s + \gamma \lambda_z; \end{cases} \quad (4)$$

the *terminal condition* $\lambda_s(T) = \mathbf{0}_n$, $\lambda_z(T) = e^{r_1 T} k N$ for costate is derived from the *transversality condition* $\lambda(T) \triangleq \Psi_x(x(T))$, where $\Psi(T) = e^{r_1 T} k z^\top(T) N$ is the *terminal cost* in the objective function $J[\cdot]$.

The interior *necessary condition of optimality* $H_u = 0$ produces a single extremal $u^*(x, t, \lambda) = \frac{\beta}{qe^{r_2 t}}(\lambda_z - \lambda_s) \odot s \odot N^{-1} \odot \mathbf{A}^\top z$, where N^{-1} is the coordinatewise inverse of the population vector, $N \odot N^{-1} = \mathbf{1}_n$. The Hessian matrix of H with respect to elements of u is trivially positive definite, $[H_{u_i, u_j}] = \text{diag}(e^{r_2 t} q N)$, thus, the control u^* indeed achieves the (local) *minimum* of $H[\cdot]$ and, therefore, optimizes $J[\cdot]$.

3.2 Numerical Solution for Two-Point Boundary Value Problem

Algorithm 1. Forward-Backward Sweep

- 1: INIT: Set initial guess of u , ▷ $u \equiv \mathbf{0}_n$
 - 2: Set convergence tolerance $\delta = 0.001$
 - 3: **while** $\|u - u_{\text{old}}\| > \delta \|u\|$ **or** $\|x - x_{\text{old}}\| > \delta \|x\|$ **or** $\|\lambda - \lambda_{\text{old}}\| > \delta \|\lambda\|$ **do**
 - 4: $x_{\text{old}} \leftarrow x$, $\lambda_{\text{old}} \leftarrow \lambda$, $u_{\text{old}} \leftarrow u$ ▷ Keep old values
 - 5: Solve $\dot{x} = f(t, x, u)$ ▷ Eq. 2
 - 6: Solve $\dot{\lambda} = g(t, x, u)$ ▷ Eq. 4
 - 7: Compute $u^*(t, x, \lambda)$
 - 8: Set $u = 0.9u_{\text{old}} + 0.1u^*$ ▷ update the control guess
 - 9: **end while**
 - 10: Done.
-

We solve the two-point boundary value problem (2),(4) numerically with the *forward-backward sweep* method (Algorithm 1), see e.g. [15, Ch. 4]. For the time interval $[0, T]$, denote the discretization points set \mathcal{T} ; we take $\mathcal{T} = \{0, \dots, T\}$, the integer points in $[0, T]$. The functions x , λ , and u are stored in the form of their values at \mathcal{T} . The norms in step 3 are maximal columnwise absolute sums.

The *initial guess* of optimal control is the “null” control $u \equiv \mathbf{0}_n$. The initial value problems for (2) and (4) at steps 5 and 6 are solved with the order 5 *adaptive step size* Dormand–Prince method—the `ode45` function [26]. Where

values *between* the points \mathcal{T} are required, for x and λ these are provided by the interpolation built into `ode45`, and the values of u are obtained through *monotone piecewise cubic* splines [11] (the `pchip` function).

A discussion of update strategies that improve the convergence can be found e.g. in [15, Sec. 8.2]; in our experiments, $u = 0.9u_{\text{old}} + 0.1u^*$ converged better than the trivial $u = u^*$ or average $u = 0.5u_{\text{old}} + 0.5u^*$.

Normalizing Node Populations for Better Scaling of λ . Populations in the objective function $J[\cdot]$ (3) add to the derivatives of costate functions $\dot{\lambda}_{z_i}$ the terms $-e^{r_1 t} c N_i$, which easily dominate these since both the control u , susceptibles s , and the connection matrix \mathbf{A} have elements in $[0, 1]$. To improve scaling, we *normalize* the population vector N to $N' \triangleq N / \max_{i \in \{1, \dots, n\}} \{N_i\}$ in costate dynamics (4) and objective function (3). On `NWtra2072`, this scaling improves `ode45`'s run time on the costate system by about 30%.

3.3 Experiment: Patient Zero in Baker County, Oregon

We consider 14 scenarios, which all run for 180 days and are meant to evoke “sheltering in place until effective treatment or vaccine become available.” The model parameters in Table 2 are adapted from [22]. The initial values are in the form of a single “patient zero” present on day 0, located in census tract 950100, Baker County, Oregon, served by the Walla Walla Regional Airport.

The solver for Algorithm 1 was executed in Matlab R2021a on a laptop with 1.4 GHz 8th-gen. Intel Core i5 CPU and 16 GB RAM under macOS 11.2.3; the visualizations were produced with the Julia interface to VegaLite declarative graphics system [25]. Experiment results are summarized in Table 3: the columns $Z^0 + R^0$ and $Z^* + R^*$ list the *cumulative infected* on day 180, J^* is the optimal objective function value, $J^0 \downarrow$ is improvement upon the null control in percent, and “Iter” is the number of iterations made before Algorithm 1 converged. Figure 3 contrasts average optimal control effort and the fraction of infected in the total population for null and optimal controls for the 2,072 census tracts in Oregon and Washington states aggregated to different levels, respectively, to state level (`NWste2`), county level (`NWcty75`), and no aggregation (`NWtra2072`).

Scenario names consist of *data set ID*, *aggregation mode*, and *node number*. Those with the same data set ID share the total population and base transportation networks. The NW2 set was designed to test Algorithm 1 on a region with about 5,000 nodes; there are 7,038 census tracts in Cal. alone, and 9,110 in the WCT set. The latter two proved to be the *saturation point* of our implementation: in the 600 sec. time limit, not even the forward dynamics system could be solved by `ode45` (step 5 in Algorithm 1), and we omit these from Table 3.

Through aggregation (see Sect. 1.3), we have been able to study several regions at different resolutions. In view of the model (1), the smaller the resolution the fewer infections because the mixing between the populations is always less than the mixing within a population. This can be observed as one sees progressively less over-estimation of infection rates as one progresses from by-state to by-AP to by-county aggregation. In addition, more fine-grained control improves the objective function.

However, the tract-level scenarios NWtra2072 and NW2tra4380 exhibit worse infection rates than even the state-level ones. We attribute this to numerical issues such as problems with automated step size selection in `ode45` and more floating point round-offs, however, more study is required to give conclusions. Let us also note that air travel is least meaningful on the *tract* level since many tracts only get fractions of a passenger per day.

Recall that these scenarios are not meant to forecast the course of an epidemic, but rather as a “visual aid” in ensuring a procedure for numerically obtaining the optimal control for dynamic system with network functions as intended, and produces meaningful results (Fig. 2).

Table 2. Model parameters

β	γ	c	q	k	r_1	r_2
0.1196	0.0437	0.002	0.00045	0.02	0.002	0.002

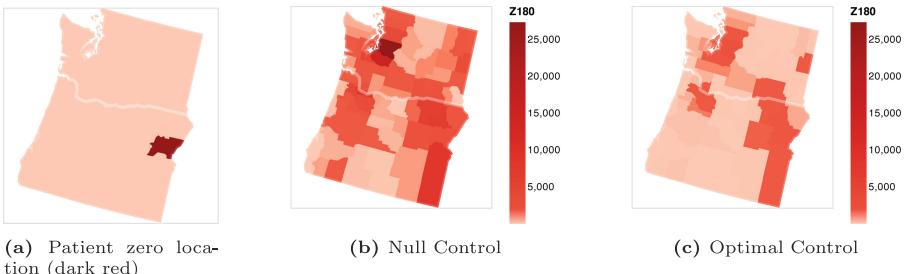


Fig. 2. Patient zero in Baker cty., Oregon; numbers of infected on day 180, NWcty75

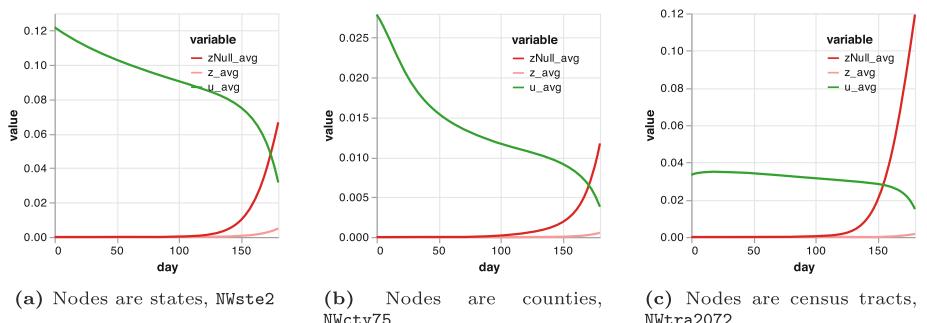


Fig. 3. Fraction of infected in the total population under optimal control (z_{avg}) and under null control (z_{Null_avg}), and average control effort (u_{avg}) in “NW” scenarios

Table 3. Forward-backward sweep for 180-day patient zero scenarios

Region	Scenario	$Z^0 + R^0$	$Z^* + R^*$	J^*	$J^0 \downarrow, \%$	Time(s)	Iter.
Ore., Wash.	NWste2	1,060,453	76,999	0.0062	20	1.01	30
	NWap23	439,769	21,961	0.0050	43	1.32	41
	NWcty75	189,027	7,827	0.0023	52	2.03	50
	NWtra2072	1,958,546	22,671	0.8078	84	37.07	65
Cal., Ore., Wash.	WCTste3	1,145,242	91,669	0.0012	12	1	30
	WCTap52	675,912	26,555	0.0040	54	1.66	43
	WCTcty133	274,434	8,997	0.0005	61	2.83	49
Contiguous U.S.	ALLste49	1,230,589	107,566	0.0015	3	1.86	31
	ALLap417	610,218	30,848	0.0028	41	4.22	44
	ALLcty3109	358,562	15,628	0.0008	47	52.75	52
Ariz., Idaho, Mont., Nev., Ore., Utah, Wash., Wyo.	NW2ste8	1,113,922	85,982	0.0069	15	0.76	31
	NW2ap74	629,166	26,371	0.0052	51	1.9	42
	NW2cty259	293,329	11,262	0.0018	56	3.85	53
	NW2tra4830	2,830,360	29,529	0.9215	85	289.19	61

4 Conclusion

We developed a procedure for generating the sets of population points and transportation networks with varying number of nodes and geographic scale, which can be used as input to optimal control problems on metapopulation epidemic models, and tested it on several scenarios covering the western U.S. We designed a simple yet meaningful *optimal control* scheme for a *metapopulation network SIR model*, and tested it on several scenarios with 2–9,110 nodes.

We found that the baseline implementation of forward-backward sweep with variable-step 5th-order ordinary differential equation solver and dense representation of the *connection matrix* could solve the 180-day scenarios on nearly 5,000 nodes in less than 300 s, yet, at 7,000 nodes or more, not even one iteration of the sweep can be done in the same time. This makes the scenarios with more than 5,000 nodes a fitting target for advanced methods that use e.g. mean field approximations [14] or graphons and mean field games [3, 9, 12].

Modeling-wise, it will be worthwhile to consider more plausible scenarios than ad hoc patient zero, and also decouple modeling resolution from control resolution e.g. by fixing the same social distancing control on a state level but still computing the model at the county level, which would net “price of uniformity” through the losses in objective function as observed in e.g. [18].

Acknowledgements. The author would like to thank Peter E. Caines, Rinel Foguen Tchuendom, and Shuang Gao for discussions on the model, data, and control. The author is especially grateful to Kara Ignatenko for help in improving the project code maintainability and the performance of the instance generator.

References

1. Achterberg, M.A., Prasse, B., Ma, L., Trajanovski, S., Kitsak, M., Van Mieghem, P.: Comparing the accuracy of several network-based COVID-19 prediction algorithms. *Int. J. Forecast.* (2020)
2. Ajelli, M., et al.: Comparing large-scale computational approaches to epidemic modeling: agent-based versus structured metapopulation models. *BMC Infect. Dis.* **10**, 190 (2010)
3. Aurell, A., Carmona, R., Dayanikh, G., Laurière, M.: Finite state graphon games with applications to epidemics. arXiv Preprint [arXiv:2106.07859](https://arxiv.org/abs/2106.07859) (2021)
4. Balcan, D., Gonçalves, B., Hu, H., Ramasco, J.J., Colizza, V., Vespignani, A.: Modeling the spatial spread of infectious diseases: The GLobal Epidemic and Mobility computational model. *J. Comput. Sci.* **1**(3), 132–145 (2010)
5. Brett, T., et al.: Detecting critical slowing down in high-dimensional epidemiological systems. *PLoS Comput. Biol.* **16**(3), 1–19 (2020)
6. U.S. Bureau of Transportation Statistics. Air carrier statistics (Form 41 traffic)—U.S. carriers. T-100 domestic market (2020). <https://www.transtats.bts.gov/>. Accessed 28 Mar 2020
7. Carli, R., Cavone, G., Epicoco, N., Scarabaggio, P., Dotoli, M.: Model predictive control to mitigate the COVID-19 outbreak in a multi-region scenario. *Ann. Rev. Control* **50**, 373–393 (2020)
8. Chao, D.L., Halloran, M.E., Obenchain, V.J., Longini, I.M., Jr.: FluTE, a publicly available stochastic influenza epidemic simulation model. *PLoS Comput. Biol.* **6**(1), 1–8 (2010)
9. Erol, S., Parise, F., Teytelboym, A.: Contagion in graphons. Available at SSRN (2020)
10. Fleming, W.H., Rishel, R.W.: Deterministic and Stochastic Optimal Control. Springer, New York (1975). <https://doi.org/10.1007/978-1-4612-6380-7>
11. Fritsch, F.N., Carlson, R.E.: Monotone piecewise cubic interpolation. *SIAM J. Numer. Anal.* **17**(2), 238–246 (1980)
12. Gao, S., Caines, P.E.: Graphon control of large-scale networks of linear systems. *IEEE Trans. Autom. Control* **65**(10), 4090–4105 (2020)
13. Kirk, D.E.: Optimal Control Theory: An Introduction. Prentice-Hall, Englewood Cliffs (1970)
14. Kiss, I.Z., Miller, J.C., Simon, P.L., et al.: Mathematics of Epidemics on Networks, vol. 598. Springer, Cham (2017). <https://doi.org/10.1007/978-3-319-50806-1>
15. Lenhart, S., Workman, J.T.: Optimal Control Applied to Biological Models. CRC Press, Boca Raton (2007)
16. Liu, F., Buss, M.: Optimal control for heterogeneous node-based information epidemics over social networks. *IEEE Trans. Control Netw. Syst.* **7**(3), 1115–1126 (2020)
17. Lovász, L.: Large Networks and Graph Limits, vol. 60. American Mathematical Society, Providence (2012)
18. Ma, Q., Liu, Y.Y., Olshevsky, A.: Optimal lockdown for pandemic control (2021)
19. Mei, W., Mohagheghi, S., Zampieri, S., Bullo, F.: On the dynamics of deterministic epidemic propagation over networks. *Ann. Rev. Control* **44**, 116–128 (2017)
20. Nowzari, C., Preciado, V.M., Pappas, G.J.: Analysis and control of epidemics: a survey of spreading processes on complex networks. *IEEE Control Syst. Mag.* **36**(1), 26–46 (2016)

21. Openflights airport database (2017). <https://openflights.org/data.html>, <https://github.com/jpatokal/openflights/data>. Accessed 21 Mar 2021
22. Ouardighi, F.E., Khmelnitsky, E., Sethi, S.: Control of an epidemic with endogenous treatment capability under popular discontent and social fatigue. Available at SSRN 3731673 (2020)
23. Paré, P.E., Beck, C.L., Başar, T.: Modeling, estimation, and analysis of epidemics over networks: an overview. *Ann. Rev. Control* **50**, 345–360 (2020)
24. Rvachev, L.A., Longini, I.M., Jr.: A mathematical model for the global spread of influenza. *Math. Biosci.* **75**(1), 3–22 (1985)
25. Satyanarayan, A., Moritz, D., Wongsuphasawat, K., Heer, J.: Vega-lite: a grammar of interactive graphics. *IEEE Trans. Vis. Comput. Graph.* **23**(1), 341–350 (2016)
26. Shampine, L.F., Reichelt, M.W.: The Matlab ODE suite. *SIAM J. Sci. Comput.* **18**(1), 1–22 (1997)
27. Vrabac, D., Shang, M., Butler, B., Pham, J., Stern, R., Paré, P.E.: Capturing the effects of transportation on the spread of COVID-19 with a multi-networked SEIR model. *IEEE Control Syst. Lett.* **6**, 103–108 (2021)



Hardness Results for Seeding Complex Contagion with Neighborhoods

Meher Chaitanya^(✉) and Ulrik Brandes

Social Networks Lab, ETH Zürich, Zürich, Switzerland
{mpindiprolu,ubrandes}@ethz.ch

Abstract. Identifying the minimum set of initiators in fixed threshold complex contagions to bring about behavioral change in a network is a well-known problem that has been studied under different names such as influence maximization or Target Set Selection (*TSS*). It is known to be hard to approximate within a polylogarithmic factor.

Recently, Guilbeault and Centola (Nature Communications, 2021) employed a novel seeding strategy in which seed nodes are included together with all of their neighbors. Referring to this variant as Neighborhood-*TSS* (*N-TSS*), we provide hardness and inapproximability results for identifying minimum-cardinality seed sets. In addition, we close a gap in the literature by extending a Strong Exponential Time Hypothesis (SETH)-based lower bound on the running time for the cardinality- k *TSS* with uniform threshold k to the case $k = 2$.

Keywords: Computational complexity · Social influence · Complex contagion

1 Introduction

Social influence is an umbrella term for multiple processes that trigger changes in individuals' behaviors due to their association with other individuals [1]. It attempts to address questions pertinent to behavioral or opinion changes, such as understanding why and when people conform or how the spread of information or behaviors occur among a group of individuals in a social structure. One particular class of models for spreading phenomena in social networks are complex contagions [2]. Unlike simple contagion, where a single influence may be enough for the transmission of information, complex contagion models the situation in which an individual may require reinforcement from multiple peers to adopt a change in behavior. The required degree of influence is modeled as a threshold associated with each vertex in the network. Examples include social movements, health behaviors, fashion trends, animal behaviors, and the spread of new technologies [2,3].

In the threshold-based models, an actor/vertex in a network can exist either in the passive state or an active state. In a passive state, a vertex can be affected by its neighbors that have adopted the behavior, and in the active state, a vertex

can influence its neighbors to adopt the behavior. The transition from passive to active state occurs when the incoming influence is at least its threshold. Thresholds in diffusion models can either be fixed [2, 4] or selected uniformly at random [5–7]. Instances of fixed threshold models include Majority Threshold (MT), Constant Threshold (CT), and Unanimous Threshold models (UT) [8, 9]. In the CT model, threshold of every vertex can be any value in between 0 and its degree in the network (the degree of a vertex is the number of edges incident on it). Both MT and UT models can be considered as special cases of Constant Threshold models wherein MT requires the threshold of every vertex i to be at least $\frac{\text{degree}(i)}{2}$ and in UT, threshold of a vertex is equal to its degree [9, 10]. It has been shown that the vertices having high closeness or betweenness centralities, etc., are not suited to spread complex contagions like politicized ideas or contentious news stories that require collective influence [11–13].

Influence maximization (*IM*) deals with identifying a small set of initiators or seed nodes in complex contagion to trigger a cascading effect. Kempe et al. [5–7] studied the diffusion models for *IM* under probabilistic thresholds by posing it as a combinatorial optimization problem. Under the assumption of choosing thresholds uniformly at random for every vertex, they show that the coverage function, σ_{cov} , that takes a set of vertices and their threshold as input and returns the cardinality of the active vertex set, is monotone and submodular. Due to submodularity property, the greedy algorithm provided by Kempe et al. [5] achieves an approximation ratio of $(1 - \frac{1}{e})$ [14]. For fixed threshold models, *IM* goes by different names such as target set selection (TSS) [9, 15], contagious sets [4, 16], r -threshold and r -monotone models [17, 18]. It has applications in many areas of viral marketing and drug abuse prevention. This problem can also be viewed as a generalization of many graph-theoretic problems such as k -tuple dominating set [19] and dynamic monopolies [20]. Also, the minimum vertex cover problem can be considered as a special case of TSS, where the threshold of every vertex in the network is equal to its degree [9]. In this case, the diffusion model for influence maximization is mostly influence-resistant. Dreyer et al. [4] showed that the *TSS* is \mathcal{NP} -hard in fixed threshold complex contagion models when threshold ≥ 3 . Chen [9] showed that for any $\epsilon > 0$ the *TSS* cannot be approximated within a factor of $\mathcal{O}(2^{\log^{1-\epsilon} n})$ unless $\mathcal{NP} \subseteq \text{DTIME}(n^{\text{polylog}(n)})$. Chen also proved that the *TSS* is \mathcal{NP} -hard when the threshold of every vertex is at most 2. Due to this, *TSS* is computationally hard both for uniform and non uniform thresholds. Many tractability results for variants of *TSS* are based on Chen's work. It can be observed that the *TSS* can be solved in linear time when the threshold of every vertex in the graph is at most 1. Recently Guilbeault et al. [13] provided seeding strategies for a variant of *TSS*, which we refer it as Neighborhood-*TSS* (*N-TSS*). They identify the influential vertices suitable for spreading fixed threshold complex contagions from a vertex and its neighborhood to any other vertex in the graph by deriving complex path length and complex centrality measures. Guilbeault et al. defined the complex path length (*CPL*) as the cardinality of the geodesic sequence of vertices encountered when the complex contagion spreads from a vertex i and its neighborhood to a vertex j .

This is described in the Eq. (1). Here $CC_{i,j}$ refers to the subgraph activated due to the spread of complex contagion from the neighborhood of i to j . The complex centrality for a vertex i , CC_i , is then defined as the average of all the complex path lengths from neighborhood of i to $v \in V \setminus i$. An actor with a high complex centrality value is designated as the influential for spreading complex contagions on the given network.

$$\begin{aligned} CPL_{i,j} &= |Geodesic(CC_{i,j})| \\ CC_i &= \frac{1}{n - \text{degree}(i)} \sum_{i \neq j} CPL_{i,j} \end{aligned} \quad (1)$$

Let G be a graph with the vertex set V and an edge set E . For convenience, we denote the cardinality of the vertex set and edge set as n and m , respectively. Let $\theta \in \mathcal{R}^n$ be a vector of thresholds where its elements, θ_i , represents the threshold of each vertex $i \in V$ with $\theta_i \in \mathbb{Z}^+$. The coverage function, σ_{cov} , takes a set of vertices and θ as its parameters and determines the amount of influence the set has on the network. The Target Set Selection (*TSS*) is formalized as follows.

Instance: A graph $G = (V, E)$ and a threshold $\theta \in \mathcal{R}^n$

Problem: Find a minimum-cardinality set $S \subseteq V$, such that $\sigma_{cov}(S, \theta) = |V|$

Output: The seed set $S \subseteq G$

For a vertex $v \in V$, let $N(v) = \{w : \{v, w\} \in E\}$ denote its neighborhood, and $N[v] = N(v) \cup \{v\}$ its closed neighborhood. We extend this to (closed) neighborhoods of sets $S \subseteq V$ via the union of their elements' (closed) neighborhoods. Neighborhood-*TSS* is formalized as follows.

Instance: A graph $G = (V, E)$ and a threshold $\theta \in \mathcal{R}^n$

Problem: Find a minimum-cardinality set $N[S] \subseteq V$, such that $\sigma_{cov}(N[S], \theta) = |V|$

Output: The seed set $S \subseteq G$.

Notice that the output of Neighborhood-*TSS* is the seed set S which together with its neighborhood (i.e. $N[S]$) forms a minimum-cardinality set that is needed to spread the behaviour in the network. The remainder of this paper is organized as follows. In Sect. 2, we will explore some properties that are useful in understanding the diffusion behavior in complex contagions. In Sect. 2.1 we will provide a lower bound for a 2-threshold complex contagion model(*TSS*) when the cardinality of seed set is 2. In Sect. 3, we will provide the hardness and inapproximability results for Neighborhood-TSS. In this paper, we will use θ and t to represent heterogeneous threshold (vector) and homogeneous threshold (scalar) respectively. We will also use the terms node and vertex interchangeably.

2 Spreading Behavior in Complex Contagions

In this section we will explore properties that are useful for understanding the diffusion behavior of complex contagions.

Lemma 1. *The Coverage function, σ_{cov} , is not supermodular.*

Proof. A function, f , is supermodular if the following equation holds.

$$f(S) + f(\mathcal{T}) \leq f(S \cup \mathcal{T}) + f(S \cap \mathcal{T})$$

Let $S, \mathcal{T} \subseteq V$ and $S \cap \mathcal{T}$ is \emptyset . Assume a symmetric contagion spread between S and \mathcal{T} , i.e., if the set S is active, it activates the set \mathcal{T} and vice versa. Let x and y be the cardinalities of the sets S and \mathcal{T} . The inequality for the coverage function, σ_{cov} , is as follows (l.h.s and r.h.s are evaluated to $2 \cdot (x+y)$ and $(x+y)$ respectively).

$$\sigma_{cov}(S, \theta) + \sigma_{cov}(\mathcal{T}, \theta) \geq \sigma_{cov}(S \cup \mathcal{T}, \theta) + \sigma_{cov}(S \cap \mathcal{T}, \theta)$$

Thus σ_{cov} not supermodular.

Observation 1. *Given a graph $G = (V, E)$ and a threshold θ , the dyadic relationship of the spread of behavior between any two vertices and their neighborhoods is not symmetric.*

Lemma 2. *For a uniform t -threshold model on a graph $G = (V, E)$, where every vertex has a threshold $t \in \mathbb{Z}^+$, the relation between minimum-cardinality seed set (OPT), vertices, and edges is given by $|OPT| \geq \frac{(t \cdot |V|) - |E|}{t}$*

Proof. Notice that for contagion to propagate from an optimal set, the number of edges must be greater than $(t \cdot (|V| - |OPT|))$ (as every vertex $v \in V \setminus OPT$ must have a degree of at least t for it to be activated and spread the contagion to its neighbors).

Observation 2. *A uniform t -threshold model, $t \in \mathbb{Z}^+$, requires a set of at least t vertices that are at most at a distance of 2 to start the spread of complex contagion.*

Observation 3. *For a θ -threshold model, $\theta \in \mathcal{R}^n$, all vertices with degree less than θ are in the minimum-cardinality seed set [15].*

In other words, these are the forced vertices that are always part of the optimal set.

Observation 4. *Let $G = (V, E)$ be a connected graph and θ be the threshold ($\theta \in \mathcal{R}^n$ and $\theta_v \geq 2, \forall v \in V$). There exists an optimal seed set for TSS in G that does not contain the vertices with an intermediate threshold of one [15].*

2.1 Lower Bound for a 2-Contagious Set

A k -contagious set is a set where the cardinality of the TSS optimal set is k . It is trivial to design a brute force algorithm that does $\binom{n}{k}$ comparisons to identify a k -contagious set. Charikar et al. [21] showed that for directed graphs, for $k \geq 3$, there exists no $\mathcal{O}(n^{k-\epsilon})$ time algorithm to identify a k -contagious

set for every $\epsilon > 0$ unless the Strong Exponential Time Hypothesis, *SETH* is false. *SETH* conjectures that for any $\epsilon > 0$, there exists an integer k such that k -*SAT* requires $\Omega(2^{(1-\epsilon)n})$ [22]. In Theorem 1, we prove the lower bound also for *2-contagious* set, i.e., we show that in directed graphs with a uniform threshold of two, there exists no $\mathcal{O}(n^{2-\epsilon})$, $\epsilon > 0$, time algorithm to identify an optimal set of cardinality 2 unless *SETH* is false (as per the observation 2, the cardinality of a TSS optimal set with uniform threshold of two is at least two).

Theorem 1. *For a uniform threshold of two, 2-Contagious set, there is no $O(n^{2-\epsilon})$ algorithm unless SETH is false.*

Proof. We reduce from CNF-SAT problem. Let F be a conjunctive normal form with n variables and m clauses, $m = O(n)$. Without loss of generality, assume 2 divides n . We build an auxiliary graph G' as follows: Partition the given n -variables of CNF-SAT formula into 2 subsets A and B , each of size $\frac{n}{2}$. For each of the 2 subsets, there are $2^{\frac{n}{2}}$ possible partial assignments. Each possible partial assignment corresponds to a node in G' . Construct a tree for each subset of partial assignment nodes in any random order. So there are two trees with $O(2^{\frac{n}{2}})$ edges in G' . There are no edges connecting a partial assignment node from one tree to another. Add a hierarchy of nodes where, in the first level, for every clause in F , we add a clause node in G' and place a directed edge from the partial assignment node to a clause node if it satisfies the clause. So we have m clause nodes in the initial level. In the second level, we have $\frac{m}{2}$ dummy nodes wherein each dummy node has an incoming edge from two nodes of the previous level, such that all nodes in the previous level have to be activated for the current level to be activated and so on. In the final level, we have a single supernode with an incoming degree of two. Add a directed edge from the supernode to each partial assignment node. Add a supporting node p and place a directed edge from p to each of the clause nodes. Add an incoming edge from every partial assignment node to p . The total number of nodes and edges in G' are $O(2^{\frac{n}{2}})$. We set the thresholds of all nodes in G' to 2. The construction of G' is depicted in Fig. 1.

We claim that the solution of the *2-contagious* set is a pair of partial assignment nodes, one from each subset that satisfies F . It can be noticed that a pair of clause nodes or a pair of dummy nodes can at most activate a single vertex in G' . Also the pair (clause node, supernode), (clause node, partial assignment node), (clause node, dummy node), (supernode, dummy node) or (clause node, supporting node) will not activate any vertex in G' .

Now consider the pair (supernode, a partial assignment node $\in A$). This pair will infect all partial assignment nodes that belongs to the set A . However, the rest of the partial assignment nodes that are in B remain inactive. Hence they together cannot form a *2-contagious* set. Similar argument can be made when the partial assignment node is chosen from the set B . The pair (a partial assignment node $\in A$, supporting node) will not be able to activate partial assignment nodes from B . The same holds when the partial assignment node is chosen from A together with the supporting node.

Two partial assignment nodes from the same subset, i.e., either from A or from B , cannot be part of the solution set as they cannot activate the entire graph G' . Hence, the only possible solution is a pair of partial assignment nodes, one from A and the other from B that together activates the supporting node and all clause nodes. These clause nodes activate the dummy nodes and supernode, which in turn activates the entire graph, G' . So, if we have an algorithm that finds a 2 -contagious set in $\mathcal{O}(|V|^c)$ for $c < 2$, then there exists a faster algorithm for solving CNF-SAT, violating *SETH*.

As a result of Theorem 1, identifying an uniform two threshold, 2 -contagious set on a graph G requires a quadratic number of comparisons.

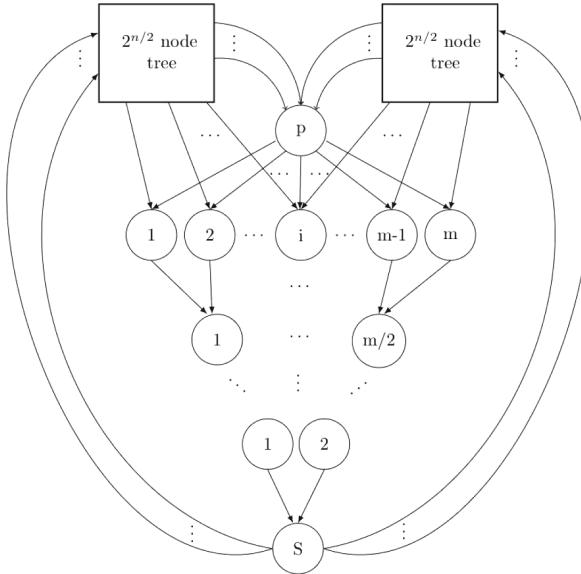


Fig. 1. Reduction from CNFSAT to 2 -contagious sets. The first level represents m clause nodes followed by a hierarchy of dummy nodes. In the final level we have a super node with an out degree of $2^{\frac{n}{2}+1}$.

3 Neighborhood-TSS

In this section, we will look at a relaxed variant of *TSS* called Neighborhood-*TSS* [13]. Rather than identifying the minimum-cardinality set of vertices to activate a graph, we turn our attention to identifying a minimum-cardinality set of vertices together with their neighborhoods to activate the entire graph. Theorem 2, Part 1, shows that it is \mathcal{NP} -hard to identify such an optimal set. Ideally, instead of considering the entire neighborhood of a vertex, we are interested in identifying a subset of neighbors having the same coverage. Theorem 2,

Part 2, shows that obtaining such a set is computationally hard. The targeted threshold model is the problem of determining if a vertex $v \in V$ and a subset of its neighborhood can together activate a vertex $u \in V$. Theorem 2, part 3 shows that identifying a minimum subset of neighbors of v to activate a vertex u is \mathcal{NP} -hard.

Theorem 2. Consider an undirected graph $G = (V, E)$. Let t be the uniform threshold of every vertex in G . Then the following decision problems are \mathcal{NP} -complete.

(Part 1) Does there exist a set of k vertices that when activated together with their neighborhoods will activate the entire graph G ?

(Part 2) Let S denote the set of vertices that gets activated with a vertex $v \in V$ and its neighborhood. Do there exist k neighbors of v that when activated together with v activate S ?

(Part 3) Given any two vertices $u, v \in V$, do there exist k neighbors of u that when activated together with u will activate v ?

Proof. For all these decision problems the reduction is from TSS [9].

(Part1) We transform the given graph G to an auxiliary graph G' as follows. We initialize the vertex set of G' to G . For every edge $(i, j) \in G$, we add a pair of dummy vertices d_i, d_j and place undirected edges $(i, d_i), (d_i, d_j)$, and (d_j, j) in G' . The total number of vertices and edges in G' are $(|V| + 2 * |E|)$ and $3|E|$, respectively. The threshold of all dummy nodes is set to one, and for the rest of the vertices in G' , it is set to t . Note that as per the construction of G' , each dummy vertex, d_i , has only two neighbors, $i \in V$ and a dummy vertex d_j . For $N-TSS$, a non-dummy vertex, say i , of degree greater than two, can always be replaced with the dummy vertex adjacent to it as i belongs to the neighborhood of d_i and activating d_i and its neighborhood will, in turn, activate the dummy neighbors of i . Thus resulting in a minimum-cardinality set comprising of $N-TSS$ seed set and their neighbors. We claim that there exists a $N-TSS$ seed set of size k in G' if and only if there exists a TSS seed set of size k in G (recall that the output of $N-TSS$ is the seed set that has the minimum-cardinality when considered together with their neighborhoods). To observe this, let S and S' denote the seed sets for TSS on G and $N-TSS$ on G' respectively. The transformation of the set S' to S is as follows. Replace any dummy vertex, $d_j \in S'$, with a non-dummy neighbor, $k \in V$, as activating k will activate d_j . The rest of the non-dummy vertices remain intact. It can be observed that this replaced set of size k activates all the vertices in G , thus providing a k cardinality TSS seed set for G . Conversely, if there exist a TSS set S of k vertices that activates G then there exist a $N-TSS$ set with k vertices in G' that together with its neighborhoods activates G' . The set S' can be constructed from the set S as follows. If a vertex $i \in (V \cap S)$ and the degree of i is at most two, add i to S' . If the degree of i is greater than two, place any one dummy vertex adjacent to i from G' to S' as this ensures the minimum-cardinality of seed set together with its neighbors to activate G' . Hence the solution to G' is linear time transferable to TSS solution on G and vice versa.

(Part2) For a given graph instance G and a threshold t we transform the graph to G' as follows. Initialize G' to G . Add an auxiliary vertex, a , to G' and place an undirected edge from a to all $v \in V \setminus \{a\}$ in G' . It can be noticed that there exists a TSS set with k vertices in G that activate the entire graph at threshold t if and only if there exist k neighbors of a that when activated together with a activates the complete set of vertices $v \in G' \setminus \{a\}$ at a uniform threshold of $t + 1$.

(Part3) For a given graph instance $G = (V, E)$ and a threshold t , we transform the graph to G' as follows. Initially set G' to G . Add an auxiliary vertex, u , to G' and place an edge from u to all $v \in V$ in G . Without loss of generality assume that $(t + 1)$ divides $|V|$. Add a hierarchy of dummy vertices as follows. In the first level we have $\frac{|V|}{t+1}$ vertices where each dummy vertex is connected to $t + 1$ unique vertices in V . In the second level, we have $\frac{|V|}{(t+1)^2}$ dummy vertices each having an edge from $t + 1$ nodes from level 1, such that all vertices from the previous level need to be activated to activate vertices in the current level and so on. In the final level, we have a single vertex w with a degree of $t + 1$. Set the threshold of all vertices in G' to $t + 1$. It can be noticed that there exists a set of k neighbors of u that together activate the vertex set V , the hierarchy of dummy nodes and finally the node w at a threshold of $t + 1$ if and only if there exists a TSS set, S , of size k in G that activates the entire graph at threshold t .

3.1 Inapproximability Results

In this section, in Lemma 3, we will show that the Neighborhood-TSS (N -TSS) is hard to approximate within a polylogarithmic factor. In Theorem 3, we will also show that N -TSS is hard to approximate better than $\mathcal{O}(\log(n))$ for bipartite graphs, unless P=NP.

Lemma 3. *N -TSS cannot be approximated within the ratio of $\mathcal{O}(2^{\log^{1-\epsilon} n})$, for any $\epsilon > 0$, unless $NP \subseteq DTIME(n^{polylog(n)})$*

Proof. The proof readily follows from Theorem 2, Part 1. If there exists a better approximation bound for N -TSS, we can apply it on the auxiliary graph, G' , to get the set S' . The same approximation ratio then carries over to get a set S for TSS, contradicting Chen's result [9].

Theorem 3. *For a bipartite graph, unless $P = NP$, there exists no polynomial time algorithm for N -TSS with an approximation bound better than $\mathcal{O}(\log(n))$.*

Proof. We perform the reduction from an instance of set cover. Let S be a class of sets, $\{S_1, \dots, S_m\}$ and U represents an universe of elements, $\{1, 2, \dots, n\}$. The minimum set cover problem deals with identifying a minimum number of sets in S to cover all elements of U . Given an instance of set cover (S, U) , we construct a bipartite graph instance, G , for N -TSS as follows. Let V_1 and V_2 be two sets of vertices in G . For each set $S_i \in S$, create a vertex S_i and add it to $V_1 \in G$. For each $j \in U$, create a vertex j and add it to $V_2 \in G$. For every i that is an element

of a set S_j , we create two subdivision vertices, $d_1^{ij} \in V_2$ and $d_2^{ij} \in V_1$, and add edges $(S_j \in V_1, d_1^{ij} \in V_2)$, $(d_1^{ij} \in V_2, d_2^{ij} \in V_1)$ and $(d_2^{ij} \in V_1, i \in V_2)$ to G . The thresholds of all subdivision vertices is set to 1. The thresholds of all other vertices, $S_i \in (S \cap V_1)$, are set to their respective cardinalities, i.e., $\theta_i = |S_i|$. The thresholds of all vertices in V_2 are set to one (it is sufficient to show this bound in non uniform threshold case as it is as hard as the uniform threshold case). As per the construction of G , activating a vertex $S_i \in V_1$ and its neighborhood (i.e., the subdivision vertices in V_2) will activate the subdivision vertices in V_1 , which in turn activates the elements of set S_i . For any two sets, S_j and S_k , if $S_k \subseteq S_j$, then activating S_j in G would activate its elements, which in turn activates the subdivision vertices associated with S_k and finally activates $S_k \in V_1$. We claim that the solution of $N\text{-TSS}$ on G is polynomial-time transferable to the solution of minimum set cover. To observe this, let opt be the optimal seed set of $N\text{-TSS}$ on G . Any subdivision vertex in opt can be replaced by a vertex $S_v \in (S \cap V_1)$ that are either at one-hop or two-hop neighborhood of it as activating S_v will in turn activate these subdivision vertices. Let us denote this replaced set by r_opt . If a vertex $v \in (U \cap V_2 \cap r_opt)$, then v can be replaced with any one of the three-hop neighbors, $S_k \in V_1$, as its threshold is one. Due to this replacement, the set r_opt contains only the vertices $S_i \in (S \cap V_1)$ and any such vertex S_i is also part of an optimal solution for minimum set cover. Conversely, the solution to the minimum set cover, S' , can be transformed to a $N\text{-TSS}$ seed set on G as follows. For any set $S_j \in S'$, if the cardinality of S_j at most two, put the vertex $S_j \in (S \cap V_1)$ in $N\text{-TSS}$ seed set. If the cardinality of $S_j \in S'$ is greater than two, place the subdivision vertex adjacent to S_j from G in $N\text{-TSS}$ seed set. Thus, an approximation factor better than $\mathcal{O}(\log(n))$ for $N\text{-TSS}$ essentially provides a better approximation ratio for set cover.

4 Conclusion

Neighborhood-TSS($N\text{-TSS}$) deals with identifying a minimum-cardinality set of vertices with their neighborhoods that together can spread a behavior to the entire graph via complex contagion. We established the hardness and inapproximability of determining a minimum-cardinality set of seeds. As a consequence, any efficiently computable index such as the complex centrality of Guilbeault et al. [13] can only lead to heuristic seeding strategies.

References

1. Friedkin, N.E., Johnsen, E.C.: Social Influence Network Theory: A Sociological Examination of Small Group Dynamics. *Structural Analysis in the Social Sciences*, Cambridge University Press (2011)
2. Centola, D.: How Behavior Spreads: The Science of Complex Contagions, vol. 3. Princeton University Press (2018)
3. Couzin, I.D.: Collective animal migration. *Curr. Biol.* **28**(17), R976–R980 (2018)

4. Dreyer, P.A., Jr., Roberts, F.S.: Irreversible k-threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion. *Discret. Appl. Math.* **157**(7), 1615–1627 (2009)
5. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146 (2003)
6. Kempe, D., Kleinberg, J., Tardos, É.: Influential nodes in a diffusion model for social networks. In: International Colloquium on Automata, Languages, and Programming, pp. 1127–1138 (2005)
7. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. *Theory Comput.* **11**(4), 105–147 (2015)
8. Valente, T.W.: Social network thresholds in the diffusion of innovations. *Social Networks* **18**(1), 69–89 (1996)
9. Chen, N.: On the approximability of influence in social networks. In: Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1029–1037 (2008)
10. Banerjee, S., Jenamani, M., Pratihar, D.K.: A survey on influence maximization in a social network. *Knowl. Inf. Syst.* **62**(9), 3417–3455 (2020)
11. Centola, D.: Influential networks. *Nat. Hum. Behav.* **3**(7), 664–665 (2019)
12. Barberá, P., Wang, N., Bonneau, R., Jost, J.T., Nagler, J., Tucker, J., González-Bailón, S.: The critical periphery in the growth of social protests. *PLoS ONE* **10**(11), e0143611 (2015)
13. Guilbeault, D., Centola, D.: Topological measures for identifying and predicting the spread of complex contagions. *Nat. Commun.* **12**, 1–9 (2021)
14. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions-I. *Math. Program.* **14**(1), 265–294 (1978)
15. Nichterlein, A., Niedermeier, R., Uhlmann, J., Weller, M.: On tractable cases of target set selection. *Soc. Netw. Anal. Min.* **3**(2), 233–256 (2013)
16. Reichman, D.: New bounds for contagious sets. *Discret. Math.* **312**(10), 1812–1814 (2012)
17. Chopin, M., Nichterlein, A., Niedermeier, R., Weller, M.: Constant thresholds can make target set selection tractable. *Theory Comput. Syst.* **55**(1), 61–83 (2014)
18. Mishra, S., Radhakrishnan, J., Sivasubramanian, S.: On the hardness of approximating minimum monopoly problems. In: International Conference on Foundations of Software Technology and Theoretical Computer Science, pp. 277–288 (2002)
19. Klasing, R., Laforest, C.: Hardness results and approximation algorithms of k-tuple domination in graphs. *Inf. Process. Lett.* **89**(2), 75–83 (2004)
20. Peleg, D.: Local majorities, coalitions and monopolies in graphs: a review. *Theoret. Comput. Sci.* **282**(2), 231–257 (2002)
21. Charikar, M., Naamad, Y., Wirth, A.: On approximating target set selection. Randomization, and Combinatorial Optimization. Algorithms and Techniques, Approximation (2016)
22. Pătrașcu, M., Williams, R.: On the possibility of faster sat algorithms. In: ACM-SIAM Symposium on Discrete Algorithms, pp. 1065–1075 (2010)



Influence Maximization in Complex Networks Through Supervised Machine Learning

Owais A. Hussain¹(✉) and Faraz Zaidi²

¹ Karachi Institute of Economics and Technology, Karachi, Pakistan

² Institute of Business Administration, Karachi, Pakistan

fzaidi@iba.edu.pk

Abstract. Identifying influential nodes in complex networks is a well studied problem in network science. Finding an optimal set of influential nodes is an NP-Hard problem and thus requires the use of heuristics to find the minimal set of nodes capable of maximizing influence in a network. Once identified, these influencer nodes can be applied in various applications such as controlling disease outbreaks, identifying infectious nodes in computer networks, and finding super spreaders for viral marketing in social networks. This paper proposes a novel approach to solve this problem by modeling it as a supervised machine learning problem. Several synthetic and real world networks with nodal and network level attributes are used to train supervised learning models. Model performance is tested against real world networks emanating from a variety of different domains. Results show that the trained models are highly accurate in identifying influential nodes in networks previously not used for training and outperform commonly used techniques in the literature.

Keywords: Influence maximization · Complex networks · Machine learning · Gradient boosting · Resilience

1 Introduction

Influence is defined as the tendency of an individual to perform an action, and trigger another individual to perform the same action. Influence maximization is the problem of identifying optimal subset of such individuals capable of maximizing influence in a social network [27]. This field has wide applications in social and other networks. Real world applications of influence maximization can be found in controlling diseases outbreaks spreading information as word-of-mouth and viral marketing processes [14] and identifying critical nodes in infrastructures such as transportation networks and power systems.

Researchers have used various approaches to solve this NP-hard problem [22]. Kleinberg and colleagues [22] suggested a greedy strategy based on submodular functions that can obtain a solution that is provably 63% within the optimal one

but does not guarantee the best solution [27]. Furthermore, greedy approaches are not scalable to moderate size networks with hundreds of thousands of nodes.

Another approach to identify influential nodes is to use heuristic methods. Widely used methods include the high-degree centrality (HD), high-degree adaptive (HDA) [18], PageRank [28], k-cores [2], betweenness centrality, closeness and eigenvector centrality [23], equal-graph-partitioning [10] and collective influence [27]. These methods have often performed indifferently on networks with varying structural properties [19, 20] and there is no single heuristic that is capable of identifying the smallest set of influential nodes.

This paper models the problem as a supervised machine learning problem where influential nodes are a function of nodal and network level attributes combined together. Using several networks, varying in size, domain and other structural properties, a set of influential nodes is identified and then machine learning models are trained to classify influential nodes in test networks. To the best of our knowledge, this is the first attempt to use supervised machine learning algorithms to identify influential nodes in complex networks.

2 Related Work

2.1 Influence Mining

Influence mining is a diverse field and many researchers have studied it in a variety of domains to: develop models for information diffusion; enhance these models; learn influence probabilities; maximize influence; observe influence propagation; apply direct data mining approaches; explore applications in real life.

[14] was the first to study the problem of identifying influential users and influence propagation in terms of data mining. They modeled a social network as Markov Random field and proposed heuristics for selection of most influential users. Extending Domingos's work, Kempe et al. [22] took greedy approximation approach to guarantee influence maximization. They focused on two basic and very frequently used time step-based diffusion models: Linear Threshold model (LT) and Independent Cascade model (IC).

Since influence maximization under both LT and IC models are NP-Complete [22], Leskovek et al. targeted the propagation problem using a different approach, calling it Outbreak Detection [24]. The outbreak detection problem is to detect the minimal set of nodes that could spread a virus in a network in a minimal time. They developed an efficient algorithm using an approach called lazy-forward optimization to select new seeds; the algorithm is 700 times more efficient than simple Greedy solution. Wei Chen et al. [9] extended IC model to incorporate propagation of negative opinions by introducing a parameter, quality factor to depict natural behaviour of people and, while maintaining Greedy method's sub-modularity property, designed a non-trivial algorithm for k seed selection that computes influence in tree-structures and further build a heuristic for influence maximization. Kazumi Saito et al. [29] discussed how to learn influence probabilities for IC model from historical influence propagation traces. They propose a method, Expectation Maximization (EM) to solve the maximization

problem by using likelihood maximization. The EM method is a heuristic, which assumes that most influential users are the ones who propagate their actions to most of their neighbours. Their work has limitations when applying to real-World problems, most significant of which is that according to their model, nodes with higher number of connections can influence more nodes and result in being influential nodes rather than being able to generate a cascading behaviour.

Goyal et al. [16] studied the same problem under an alternate model, the General Threshold Model. They used this model and made the probabilities decline with time. The assumption behind this is that if a user u performs an action, then his linked user v will either repeat the action shortly after, or never. This decline of probability is observed to be exponential. They compared two genres of models; one assumes the influence probability remains static with time; 3 models, Bernoulli distribution, Jaccard index and Partial Credits and their combinations were discussed. The contrary assumes the influence probability to be a continuous function of time. The results showed that time-aware model was better, not only in terms of accuracy but being able to roughly predict future action times.

Lu et al. [25] comprehensively summarized the area of influence mining on different types of networks, discussing centrality methods to state-of-art heuristics to Greedy algorithm and its variations, the authors conclude on the applications of influence mining in social networks, financial area, scientific influence.

2.2 Machine Learning

Surprising as it may seem, there is scarcity of studies which utilize the power of continuously evolving machine learning techniques in order to identify influential nodes in a network in a generic fashion. Among those that exist, the prominent publications are often specific to social networks.

[17] demonstrate how predictive analytics can be used to predict diffusion cascades in a social network. Their proposed model, T-BaSIC does not assume a fixed diffusion probability, but a time-dependent function. This temporal diffusion model is used to create time series to describe how a topic originated and spread in a Twitter network in a closed set of users. The authors trained multiple classifiers to learn parameters of diffusion function, and opted Bayesian classifier to define the diffusion function. The results show that in the information diffusion process, initially the volume of tweets is high against a topic, and then lowers with time following a wave pattern. The T-BaSIC method reduces the overall diffusion prediction error by 32.75% in comparison with 1-time lag model.

Zanin et al. [33] discusses the application of data mining techniques in complex networks. The authors describe how data mining techniques can make use of structural patterns of individual nodes as well as whole network. Excerpts from this article are widely reflected in experiments we propose in this manuscript.

Albeit these attempts, to the best of our knowledge there is lack of any published work which solves the influence mining problem using supervised machine learning on a variety of complex networks.

3 Design and Methodology

The experiments discussed in this paper consist of openly available real graphs as well as synthetically generated graphs. Moreover, unlike majority of studies, which use network data, user attributes, action logs and other information, we only use network's structural properties that can be calculated by $G = (V, E)$, where V is a set of vertices or nodes in the graph G , while E is the set of edges or links connecting these vertices. Source code of all experiments conducted in the study are shared on GitHub¹.

3.1 Training Data Sets

We used networks from various domains with different sizes and structural properties. A total of 390 graphs were used to train and validate the model. The networks used in this study come from three sources:

Synthetic Networks are generated networks, which exhibit similar properties as real networks. We drew 6 random samples for each of the following types of complex networks using Igraph library ranging from size 10 - 2000 nodes [12]:

1. Scale free: almost all complex networks are scale free [3]. They are defined by a prominent property that their degree distribution obeys power-law². The power law coefficient in a scale free network is usually $2 < \gamma < 3$.
2. Small-world: these networks have low APL and high CC [31].
3. Small-world and scale free: most of the real world networks, especially social networks exhibit both scale free and small-world properties [18]

Cited Networks are frequently cited networks from various domains and sizes that are available online:

1. Zachary karate: members of a university karate club by Wayne Zachary.
2. World trade: data about manufacturers of metal among 80 countries [30].
3. Nematode: Neural network of the nematode [32].
4. Political blog: hyperlinks between weblogs on US politics [1].
5. Yeast protein: protein-protein interaction network of yeast [21].

Extracted Networks additionally, we extracted two social networks to ensure that the training set not only contains clean and ready-to-use networks but also replicate real-world situations where networks are in raw state without any preprocessing. These networks are described as follows:

¹ <https://github.com/seekme94/influence-mining/tree/influential-node-prediction/Experiments>.

² States that a change in one quantity X results in a proportional relative change in another quantity Y .

1. Influence citations: network of paper citations on topic of influence mining till 2016. The data set was constructed by collecting a corpus of research articles on various topics of Influence mining. Each node in this network represents a unique article, while the edges represent citations between these articles.
2. Twitter: starting with a reference user account (ID: seekme_94) in the Twitter network, a subset network was extracted, such that each node is either a direct connection of reference account, or a connection at a distance of 1 degree. The reference node was finally removed from the network.

3.2 Test Data Sets

To test the final model, following complex networks from various domains were used. Again, variation in their nodal as well as structural properties can be observed in Table 1. All these are also frequently cited networks in the literature. These networks can be downloaded from KONECT: <http://www.konect.cc>.

- Author NetScience: coauthorship network of network theory researchers.
- ITA 2000: air transport network representing connections between airports through direct flights.
- AS-CAIDA: network of autonomous systems of the Internet connected with each other from the CAIDA project.
- JDK Dependencies: a network of class dependencies in JDK v1.6.

Table 1. Networks used for Training. Basic statistics for all the real networks used for training the model. APL is Average Path Length, γ is Power-law coefficient, CC is clustering coefficient and AvDg is Average Degree. The variation in size of the networks, APL, γ , CC and AvDg demonstrates how we have captured different network level structural properties for our training dataset.

Network	Nodes	Edges	APL	γ	CC	AvDg
Training data						
Karate club	34	78	2.4	2.12	0.256	4.6
World trade	80	1000	1.7	2.56	0.460	25
Nematode	297	2359	2.5	3.29	0.181	15.9
Political blog	1222	19089	2.7	3.89	0.173	4.6
Yeast protein	1458	3941	6.1	3.24	0.226	31.2
Influence citation	1254	1993	3.8	2.35	0.057	3.2
Twitter	894	2060	4.9	4.60	0.052	5.4
Test data						
Author	379	914	6.0	3.36	0.431	4.8
ITA 2000	3304	67195	4.0	1.69	0.249	40.7
AS-CAIDA	26475	53381	3.9	2.09	0.007	4.0
JDK	6434	150985	2.1	2.01	0.011	46.9

3.3 Influential Node Identification

The first step to train supervised machine learning models is to identify influential nodes in training dataset.

Influence Maximization Algorithms. In order to guarantee that the nodes we label as influential, in fact are influential, we used the optimal solution (NP-hard) for networks of size less than 50 nodes, and baseline Greedy algorithm for larger networks [22]. Opting for greedy algorithm was a compulsion because finding the optimal solution for larger graphs is practically impossible in polynomial time.

Influence Test. In order to quantify the influence of a node, multiple methods exist, including diffusion test under linear threshold model, diffusion test under independent cascade model [22] and resilience test [11] which quantifies the network breakdown after a set of nodes is removed. Since the resilience test does not require any data other than network information, therefore this study uses resilience test as many other similar studies [18, 27].

Conventionally, the resilience test is a temporal function, i.e. in discrete time step, one node is removed from the network and we compute the largest connected component remaining in the network. This iteration is repeated until network completely decomposes. The quicker a method decomposes the network, the better its performance. In this article, instead of single node, a batch of k nodes is removed in exactly one iteration and the size of the biggest connected component is measured. The smaller the size of the biggest connected component, the better the identification of influential nodes.

Budget. Selecting the right budget is a widely debated topic. Pareto principle [26] suggests that the budget should be 20%, while [22] argue that this number is much smaller. Leskovec et al. [24] showed that the growth in gain with respect to budget is logarithmic. Nevertheless, this is a debatable topic and out of scope of this study. In the experiments, this budget was fixed to 10% for networks with nodes less than 50 and, 2.5% for all other networks.

3.4 Structural and Nodal Attributes

For a machine learning model to perform optimally, feature selection and data transformation are the key steps. For this experiment, we carefully chose a number of nodal and network level traits which are described in Table 2.

Table 2. Network traits, and heuristics used in the experiments.

Name	Description
Structural attributes	
Nodes	Number of nodes in the network
Edges	Number of edges in the network
Maximum degree	Maximum number of connections of a single node
Average degree	Average number of edges for each node
APL	Average of lengths of shortest path from all pairs of nodes
CC	Connectedness of nodes in the network. Value of 1 means the network is a clique
Diameter	Shortest distance between the two farthest nodes
Density	Ratio between existing network edges and all possible edges among nodes
Average distance	Average distance when connecting two nodes
Triads	Number of triangles, i.e. 3 interconnected nodes
Assortativity	Pearson correlation coefficient of nodes at any side of an edge. Ranging between -1 and 1, negative value means that the high degree nodes tend to connect with low degree nodes, positive value means that the nodes connect with respect to degree similarity
Nodal attributes	
Degree	Number of direct connections of a node to other nodes
Betweenness	The number of times a node falls along the shortest path between two other nodes
Closeness	Reciprocal of the sum of length of shortest paths between a node with all other nodes
Eigenvector	Ranks node u based on not only the degree, but also the degrees of the v nodes that are direct neighbours of u
Eccentricity	The reciprocal of the longest shortest path between a node u and all other nodes in the network v the shortest path between the node v and all other nodes in the network
Heuristics	
Coreness	Defines which core does a node lie in. A k -core of is a maximal subgraph in which each node has at least k degree. If a node belongs to k -core, but not $(k+1)$ core, then its coreness is k . This heuristic is chosen due to its performance as claimed by [2]. [4] computes coreness in $O(m)$ time.
Pagerank	Eigenvector centrality based Webpage ranking algorithm developed by Google Inc. [28]. Pagerank of a node is high if the sum of ranks of its neighbours is also high. This algorithm has proven to be effective on scale
Collective influence	A state-of-art influence maximization algorithm by Morone et al. [27] (see 2). This algorithm identifies weak nodes, which act as bridges between strong influences, and is also scalable to large networks

4 Experiments and Results

4.1 Machine Learning Models

Fernandez et al. [15] argued that the real world classification problems do not require too many machine learning models after comparing 179 classifiers on 121 data sets. Analyzing the results of the best and worst performing models from different families, we used the following models for our experiment:

1. Logistic regression: Generalized Linear Model (**GLM**) by Dobson [13].
2. Decision Trees and Rule-Based: Recursive partitioning (**Rpart**) by [6], and **C5.0**, (an extended version of original C4.5 model).
3. Support Vector Machines: (**SVM**) model using Gaussian kernel from popular LibSVM library [7].
4. Boosting: Random Forests (**RForest**) by [5], and eXtreme Gradient model by [8] (**XGBoost**).

From the dataset, the target variable tells whether the node is identified as an influential node by the Optimal or Greedy algorithm.

The models are evaluated using measure of accuracy, a standard way to compare the performance of machine learning models. We denote true positives as the number of nodes which were correctly classified as influential; true negatives is the number of nodes which were correctly classified as non-influential; false positive is the number of nodes which were incorrectly classified as influential; finally, false negative is the number of nodes which were incorrectly classified as non-influential. We test the accuracy of the models on various sizes of networks and average out the accuracy of each model on the validation set.

Figure 1 illustrate the accuracy of classification models by size, where overall accuracy is given in Fig. 2. Size of the networks are on horizontal axis, while the accuracy of each model on these sizes are on vertical axis. It is evident that XGBoost has the highest accuracy overall on varying sizes of networks. We also observe that the performance of all models in small graphs is variable. However, XGBoost still outperformed other models on small networks. Therefore, we selected XGBoost based on its performance in comparison with other supervised machine learning models. However, it is to note that the variance in performances of these models is not high. This observation also implies that the application of machine learning classification is not subject to a specific model, but with a small percentage trade-off a broad range of models can be applied to the underlying problem.

4.2 Results

The key objective of this study is to find out if a machine learning algorithm can outperform heuristics in discovering influential nodes in networks of various sizes and structural properties. Previous section explained the methodology followed in order to select and train an appropriate model, i.e. XGBoost model. In order to test our hypothesis, we pass all the test networks discussed in Table 1 to the

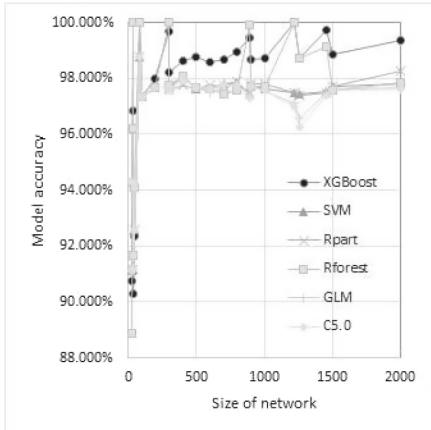


Fig. 1. Accuracy of classification models on various sizes of networks in validation set. The figure demonstrates that the models perform consistently on a variety of sizes.

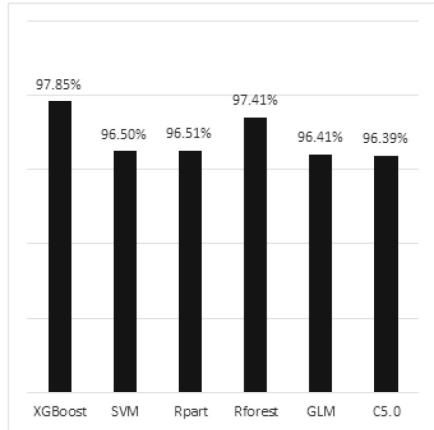


Fig. 2. Average accuracy of classification models on all networks in validation set. The results from previous figure are averaged out for all sizes.

model to identify k most influential nodes. The output obtained from the model is a probability against each node, which quantifies how strongly the model classified that particular node as influential. We pick the set of k nodes of the highest probabilities and perform resilience test on the test networks. Resilience test is destructive, i.e. it is used to measure how much a network decomposed when a set of nodes is removed. After removing k influential nodes identified by the model, we calculate the size of the largest connected component in the remaining network. Therefore the method, which identifies k nodes that results in higher decomposition of the network qualifies as a better performer.

The results shown in Fig. 3 illustrate that the XGBoost model performed very well when compared with commonly used heuristics. In this figure, “size” on x-axis represents the number of nodes in the original test network, while the rest of the bars represent how large the network remained after removing k influential nodes. Best results are highlighted with bold-italic text in the figure.

1. **Author:** this is the smallest of all test networks. XGBoost model collapsed the network to 254 after removing k influential nodes.
2. **ITA 2000:** XGBoost showed best results with network size dropping to 3014, but only marginally better than Pagerank and Betweenness centrality.
3. **AS-CAIDA:** a relatively large network, where XGBoost (score 16264) dominated other methods with significant margin. Eccentricity (score 22756), which is the next best is still had 6492 nodes.
4. **JDK:** Similar results with XGBoost (score 4696). Interestingly, Eigenvector centrality and Eccentricity are only slightly better than random method, indicating that centrality based methods may not universally perform better.

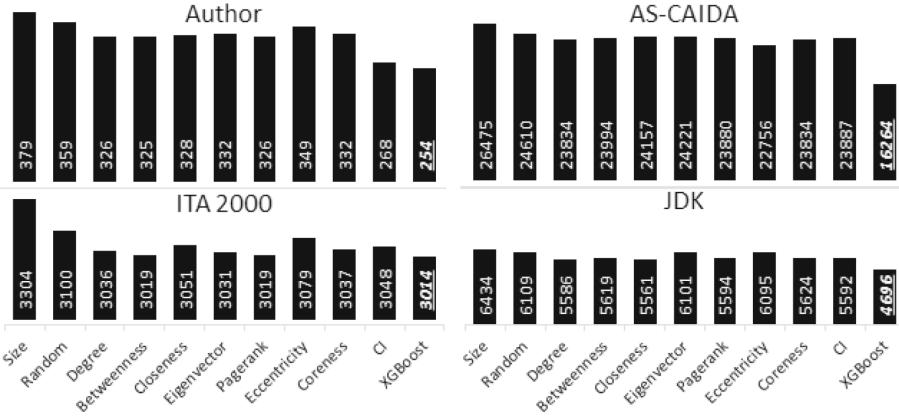


Fig. 3. Comparison of influence mining performance. On x-axis are the influence mining methods under comparison. On y-axis is the number of nodes in largest component after removing k nodes from the network

Table 3. Comparison of heuristics with ML model. XGBoost outperformed all other heuristics, while in the other results, there is no clear second best method.

Network	Author	ITA 2000	AS-CAIDA	JDK
Random	10	10	10	10
Degree	4.5	5	3.5	3
Betweenness	3	2.5	7	6
Closeness	6	8	8	2
Eigenvector	7.5	4	9	9
Pagerank	4.5	2.5	5	5
Eccentricity	9	9	2	8
Coreness	7.5	6	3.5	7
CI	2	7	6	4
XGBoost	1	1	1	1

Analyzing the results, although XGBoost model stands atop, analysis on next best also reveals that there is no clear runner up method. Table 3 lists the influence methods and ranks using average ranking method. As observable, CI has 2nd rank in Author network, but fails on other networks; Betweenness and Pagerank perform equally well on ITA 2000, but do not hold this position for other networks; likewise, Eccentricity is impressive on AS-CAIDA network, but performs almost similar to random method on other networks. This observation also validates that heuristics perform differently, based on structure of the network. Therefore, no heuristic can be considered as a generic algorithm to accurately identify influential nodes in a variety of complex networks.

Another observation is that not all networks decompose with the same rate. For example, ITA 2000 network proved highly resilient by sustaining a size of 3014 (91%) after removing influential nodes. On the other hand, AS-CAIDA reduced to 16264 (61%). It is also evident that random methods perform poorly, as indicated by various authors before.

5 Conclusion and Future Work

This study proposed use of supervised machine learning to generalize the problem of influence mining in complex networks of different structural properties. The experiments and results not only validate the proposal by outperforming various conventional and state-of-art methods, but take us a step ahead in terms of prerequisites. The model which was trained for this study was based on synthetic networks, combined with some openly available network data sets. Moreover, it did not require large amount of data and high performance computing machines. The classification outperformed the competing methods in all tested real world networks. While machine learning can certainly help in various applications, it is important to note that training the model is computationally much more expensive than any other heuristic in comparison. It is assumed that anyone applying this solution aims to reuse the model on various networks, and multiple times. For simple one-time applications, the solution is still valid in terms of performance, but has a high computation overhead. Therefore, one of the key areas of optimization is selecting the right set of features which reduces the overall complexity while maintaining the high performance achieved. This study opens up many potential research avenues to explore, and help develop robust methods to identify influential nodes in large complex networks.

References

1. Adamic, L.A., Glance, N.: The political blogosphere and the 2004 u.s. election: divided they blog. In: LinkKDD '05 Proceedings, pp. 36–43. ACM Press (2005)
2. Bae, J., Kim, S.: Identifying and ranking influential spreaders in complex networks by neighborhood coreness. *Physica A: Stat. Mechanics* **395**, 549–559 (2014)
3. Barabási, A.L., Albert, R.: Emergence of Scaling in Random Networks. *Science*, pp. 1–11 (1999)
4. Batagelj, V., Zaversnik, M.: An O(m) Algorithm for Cores Decomposition of Networks. *arXiv* 1(49), 1–10 (2003)
5. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
6. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. CRC Press (1984)
7. Chang, C.C., Lin, C.J., Chang, C.-C., Lin, C.J.: LIBSVM: a library for support vector machines. *ACM (TIST)* **2**(3), 27 (2011)
8. Chen, T., Guestrin, C.: Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference, pp. 785–794 (2016)
9. Chen, W., Collins, A.: Influence maximization in social networks when negative opinions may emerge and propagate. In: SDM 2011 (2011)

10. Chen, Y., Paul, G., Havlin, S., Liljeros, F., Stanley, H.E.: Finding a better immunization strategy. *Phys. Rev. Lett.* **101**(5), 058701 (2008)
11. Cohen, R., Erez, K., Ben-Avraham, D., Havlin, S.: Resilience of the internet to random breakdowns. *Phys. Rev. Lett.* **85**(21), 4626 (2000)
12. Csardi, G., Nepusz, T.: The igraph software package for complex network research. *Int. J. Complex Syst.* **1695** (2006). <http://igraph.sf.net>
13. Dobson, A.J.: An intro. to generalized linear models. CRC Press (2018)
14. Domingos, P., Richardson, M.: Mining the Network Value of Customers. In: Proceedings of the Seventh ACM SIGKDD International Conference, pp. 57–66 (2001)
15. Fernández-Delgado, M.: Do we need hundreds of classifiers to solve real world classification problems? *J. ML Res.* **15**(1), 3133–3181 (2014)
16. Goyal, A., Bonchi, F., Lakshmanan, L.: Learning influence probabilities in social networks. In: 3rd ACM Conference - WSDM 2010, p. 241. ACM Press (2010)
17. Guille, A., Hacid, H., Favre, C.: Predicting the temporal dynamics of information diffusion in social networks. In: Proceedings of the 21st International Conference on WWW (2013)
18. Holme, P., Kim, B.J., Yoon, C.N., Han, S.K.: Attack vulnerability of complex networks. *Phys. Rev. E* **65**(5), 056109 (2002)
19. Hussain, O., Zaidi, F., Rozenblat, C.: Analyzing diversity, strength and centrality of cities using networks of multinational firms. *Netw. Spat. Econ.* **19**(3), 791–817 (2019)
20. Hussain, O.A., Zaidi, F.: Empirical analysis of seed selection criterion for different classes of networks. In: IEEE - SCA 2013, pp. 348–353 (2013)
21. Jeong, H., Mason, S.P., Barabási, A.L., Oltvai, Z.N.: Lethality and centrality in protein networks. *Nature* **411**(6833), 41–42 (2001)
22. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: 9th ACM SIGKDD International Conference, p. 137. ACM Press (2003)
23. Landherr, A., Friedl, B., Heidemann, J.: A critical review of centrality measures in social networks. *Bus. Inf. Syst. Eng.* **2**(6), 371–385 (2010)
24. Leskovec, J., Krause, A., Guestrin, C.: Cost-effective outbreak detection in networks. In: 13th ACM SIGKDD International Conference, pp. 420–429. ACM Press (2007)
25. Lü, L., Chen, D., Ren, X.L., Zhang, Q.M., Zhang, Y.C., Zhou, T.: Vital nodes identification in complex networks. *Phys. Rep.* **650**, 1–63 (2016)
26. Mornati, F.: Manuale di economia politica, vol. 40. Studio Tesi (1971)
27. Morone, F., Makse, H.a.: Influence maximization in complex networks through optimal percolation: supplementary information. *Current Sci.* **93**(1), 17–19 (2015)
28. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Tech. rep, Stanford InfoLab (1999)
29. Saito, K., Nakano, R., Kimura, M.: Prediction of information diffusion probabilities for independent cascade model. *Lecture Notes in CS* **519**, 67–75 (2008)
30. Smith, D.A., White, D.R.: Structure and dynamics of the global economy: network analysis of international trade 1965–1980. *Soc. Forces* **70**(4), 857–893 (1992)
31. Watts, D.J., Strogatz, S.H.: Collective dynamics of a small-world' networks. *Nature* **393**(June), 440–442 (1998)
32. White, J.G., et al. E.S.: The structure of the nervous system of the nematode *c. elegans*. *Phil. Trans. R. Soc. London, B: Bio. Sci.* **314**, 1–340 (1986)
33. Zanin, M., Papo, D.: Combining complex networks and data mining: Why and how. *Phys. Rep.* **635**, 1–44 (2016)



Story of Two Populations in Epidemics: Is Every Infection Counted?

Van Sy Mai¹ and Richard J. La²(✉)

¹ ANTD, NIST, Gaithersburg, MD 20899, USA
vansy.mai@nist.gov

² ACMD, NIST, Gaithersburg, MD 20899, USA
richard.la@nist.gov

Abstract. Many previous studies on epidemic processes assume that most infected nodes that contribute to the spread of infection can be identified and accounted for. But, in some cases, this assumption may be invalid and many infection cases may go undetected. For example, the operators of subsystems in complex systems may be unaware that their systems are compromised by malware or a virus, which may go undetected for a long period. Similarly, an infectious disease can cause widely varying symptoms and some infected individuals may exhibit little to no symptoms. In these scenarios, it would be difficult to identify all infected systems or individuals, which can play a critical role in spreading the infection by malware, virus or disease. For this reason, it is of interest to devise a means of quickly determining the presence of such undetected infection cases in a network. We propose a simple optimization-based approach that can be used to determine whether or not a significant fraction of infection cases are undetected and thus are missing in reported statistics. We present numerical results obtained in a case study using publicly available COVID-19 data from four countries.

1 Introduction

The spread of malware and viruses in computer networks and information systems, infectious diseases, rumors or new product information is often studied using epidemic processes [6, 9, 10]. Many, if not most, of these studies assume that (most) infected nodes can be identified so that the parameters of the epidemic processes, such as reproduction numbers, can be estimated from available data. In some cases, however, such information may not be available or cannot be obtained easily. For example, when subsystems in a network with inadequate security measures are infected by malware or a virus, network managers may be unaware of the infection [7]. Similarly, an infectious disease can cause widely varying symptoms in infected individuals and some individuals may exhibit no obvious symptoms that can be easily identified [13].

Preventing computer malware/viruses or an infectious disease from causing a widespread outbreak requires reliable information about its transmission dynamics at its onset. This task becomes more challenging when it is difficult to get a

clear picture of the underlying epidemic process in a network or society; missing a significant fraction of infection cases in reported numbers at an early stage of an outbreak would likely lead to a costly underestimation of the severity of the problem at a crucial point. This may delay urgent measures and policies that should be put in place to curb the spread.

We propose a new approach to determining whether reported numbers of infections are accurate or there are many unreported cases. Our approach borrows insights from (statistical) model selection [11] and is based on a simple observation that infected devices or individuals that go undetected spread the infection differently than those that are detected shortly after infection and likely benefit from remediation (Sect. 2). A key challenge is that the proposed method must make a determination based solely on the available noisy information.

For our case study, we apply our approach to publicly available COVID-19 data (Sect. 5). We approximate the time-varying transmissibility of infected individuals with the help of parameterized gamma distributions. We compare the errors between (i) the “best” model with two populations (*detectable* and *undetectable* nodes) and (ii) the “best” model only with a single population (*detectable* nodes). We declare that there is a non-negligible fraction of undetectable infections if (a) the error achieved by the former is considerably less than that of the latter and (b) the estimate of the fraction of unreported infection cases is non-negligible. Numerical results we obtained using the real data for four countries that experienced a COVID-19 outbreak are presented in Sect. 5.

A Few Words on Notation: the natural number system, the nonnegative real number system, and the positive real number system are denoted by $\mathbb{N} := \{1, 2, \dots\}$, $\mathbb{R}_+ := [0, \infty)$, and $\mathbb{R}_{++} := (0, \infty)$, respectively. We will use boldface letters or symbols to denote vectors.

2 Setup

As mentioned earlier, our goal is to quickly determine, on the basis of a time series consisting of the numbers of confirmed reported infection cases over a period, whether the reported numbers accurately reflect the true numbers or they are missing many undetected, hence unreported infection cases. In order to make progress, we adopt the well-known susceptible-infected-recovered (SIR) model to capture the transmission dynamics.

The system or population consists of P (nondescript) subsystems or individuals, which we refer to as *nodes*. We assume that P is large and denote the set of P nodes by \mathcal{V} . The set \mathcal{V} can be partitioned into two populations – *detectable* or *undetectable*: when a detectable node (DN) is infected, it exhibits symptoms associated with the infection and is identified as ‘infected’. By contrast, when an undetectable node (UN) becomes infected, it goes undetected and thus is not identified as ‘infected’. For example, when a computer or device is infected with a virus, it may run a malicious process in the background or conduct a network probe, looking for vulnerabilities. Such activities can be detected by appropriate

network monitoring and analysis tools in a suitably protected network. On the other hand, a poorly monitored network, which represents an UN, may not have the capability to detect an infection, which will go uninvestigated as a result. In the case of an infectious disease, a DN is an individual who, when infected, will exhibit identifiable symptoms and be reported as a confirmed case. In contrast, an infected UN is either asymptomatic or does not manifest her infection with serious symptoms, and hence is not reported as a confirmed case.

Let $p_U \in [0, 1)$ be the fraction of undetectable nodes in \mathcal{V} , which is unknown beforehand, and define $P_U := P \cdot p_U$ and $P_D := P - P_U = P(1 - p_U)$. We aim to devise a method for determining whether $p_U \approx 0$ or not. Our approach is based on the following observation: when DNs are infected and exhibit symptoms, they will be identified and isolated in order to stop the spread of infection. As a result, most of the transmissions will take place before its infection is detected. For instance, according to the U.S. Centers for Disease Control and Prevention (CDC), approximately 50% of COVID-19 transmissions occur prior to the onset of symptoms (presymptomatic) [1]. On the other hand, when UNs are infected, because they are unaware of the infection, they continue to spread the infection to others over longer periods. We take advantage of this discrepancy in the manner in which DNs and UNs spread infection to devise our scheme.

3 Approach and Model

The key aspect of the discrepancy in the spread of infection by DNs and UNs, which we hope to exploit in our approach, can come in different forms. For example, when we consider the spread of malware/viruses in a network, even when they are infected, the devices equipped with the latest anti-malware/virus software will likely detect the infection sooner than devices with inadequate protection. For this reason, their infection periods will vary widely. In the case of an infectious disease, infected individuals with more pronounced symptoms will likely be isolated or hospitalized, preventing them from further spreading the disease, whereas infected yet asymptomatic individuals will continue to spread the disease, possibly with lower infectiousness.

This observation tells us that in the event that a DN becomes infected and its infection is detected, it will be removed from the network so that its infection does not spread and is put back in the network only after it is no longer infectious upon recovery/repair. Therefore, once detected, even though the infected DN may remain infectious, its ability to transmit the infection will be limited. For this reason, we find it convenient to differentiate the rate at which infected nodes can spread its infection from its infectiousness and call it the *transmissibility*.

3.1 Transmissibility Function

Following an approach similar to [8], we approximate the time-varying transmissibility of an infected node with the help of some function: suppose that a DN (resp. UN) is exposed to the etiological agent at time $t_0 \in \mathbb{R}_+$. Then,

the transmissibility of the infected node at time $t \geq t_0$ is given by $f_D(t - t_0)$ (resp. $f_U(t - t_0)$). Moreover, the *total* transmissibility over time, which is given by $\int_{\mathbf{R}_+} f_D(\tau) d\tau$ or $\int_{\mathbf{R}_+} f_U(\tau) d\tau$, is related to the expected number of transmissions or the (basic) reproduction number of an infected DN or UN, respectively [8].

3.2 Discrete-Time Approximation of Transmissibility Function

For our study, we adopt a discrete-time model and approximate the evolution of the total number of infection cases using the following equations, which can be viewed as a discrete-time approximation of the von Foerster equation used in [8] with two populations – DNs and UNs: for each $n \in \mathbb{N}$, let $N_D(n)$ (resp. $N_U(n)$) be the number of DNs (resp. UNs) newly infected at time $n \in \mathbb{N}$ and $\mathbf{N}(n) := (N_D(n), N_U(n))$. Similarly, let $I_D(n) := \sum_{k=1}^n N_D(k)$ (resp. $I_U(n) := \sum_{k=1}^n N_U(k)$) be the total number of infected DNs (resp. UNs) up to and including time n , and $\mathbf{I}(n) := (I_D(n), I_U(n))$. For each $n \in \mathbb{N}$, define $I_T(n) := I_D(n) + I_U(n)$, $q(n) := I_T(n)/P$, $q_D(n) := I_D(n)/P_D$, and $q_U(n) := I_U(n)/P_U$.

Suppose that there is finite $d_{\max} \in \mathbb{N}$ such that the discrete-time transmissibility functions $\mathcal{K}_D(k)$ and $\mathcal{K}_U(k)$ for an infected DN and UN, respectively, are negligible for $k > d_{\max}$. Under this assumption, we can view the discrete-time transmissibility functions as finite sequences $\mathcal{K}_D = (K_D(n) : n = 1, 2, \dots, d_{\max})$ and $\mathcal{K}_U = (K_U(n) : n = 1, 2, \dots, d_{\max})$. For notational convenience, for each $n \in \mathbb{N}$, we denote the finite sequences $(N_D(n') : n' = n - d_{\max} + 1, \dots, n)$ and $(N_U(n') : n' = n - d_{\max} + 1, \dots, n)$ by $\mathcal{N}_D(n)$ and $\mathcal{N}_U(n)$, respectively.

The aggregate transmissibility of the previously infected nodes at time n is equal to

$$N_T(n) = \mathcal{N}_D(n-1) \circledast \mathcal{K}_D + \mathcal{N}_U(n-1) \circledast \mathcal{K}_U, \quad (1)$$

where $\mathcal{N}_D(n-1) \circledast \mathcal{K}_D := \sum_{k=1}^{d_{\max}} N_D(n-k) K_D(k)$ and $\mathcal{N}_U(n-1) \circledast \mathcal{K}_U := \sum_{k=1}^{d_{\max}} N_U(n-k) K_U(k)$. Using the aggregate transmissibility, we can compute the expected numbers of newly infected DNs and UNs at time n given by

$$N_D(n) = N_T(n) (1 - p_U) (1 - q_D(n-1)) \quad \text{and} \quad (2a)$$

$$N_U(n) = N_T(n) p_U (1 - q_U(n-1)). \quad (2b)$$

Recall that $q_D(n-1) = I_D(n-1)/P_D$ and $q_U(n-1) = I_U(n-1)/P_U$. Substituting these in (2a) and (2b) and taking their ratio tells us $N_U(n)/N_D(n) = (P_U - I_U(n-1))/(P_D - I_D(n-1)) = (P_U - I_U(n))/(P_D - I_D(n)) = N_U(n+1)/N_D(n+1)$ for all $n \in \mathbb{N}$ and, thus, $I_U(n)/I_D(n)$ is constant. In view of this observation, it is reasonable to expect the ratio $I_U(n)/I_D(n)$ to remain close to $p_U/(1 - p_U)$, and we make a simplifying assumption $q_D(n) = q_U(n) = q(n)$ for all $n \in \mathbb{N}$.

This assumption allows us to simplify Eqs. (1) and (2a) as follows.

$$\begin{aligned} N_T(n) &= \mathcal{N}_D(n-1) \circledast \left(\mathcal{K}_D + \frac{p_U}{1-p_U} \mathcal{K}_U \right) \quad \text{and} \\ N_D(n) &= \mathcal{N}_D(n-1) \circledast (((1 - p_U) \mathcal{K}_D + p_U \mathcal{K}_U)(1 - q(n-1))) \end{aligned} \quad (3)$$

Thus, if we knew p_U and the transmissibility functions \mathcal{K}_D and \mathcal{K}_U , we would be able to predict the evolution of $N_D(n)$ and $N_U(n)$, $n \in \mathbb{N}$. This suggests that we can approach our problem from the perspective of *model selection*, where we wish to identify the functions \mathcal{K}_D and \mathcal{K}_U as well as the parameter p_U .

Unfortunately, without imposing additional assumptions, accurate estimation of the parameter p_U and the functions \mathcal{K}_D and \mathcal{K}_U requires a large number of samples, which may not be available, especially at the onset of an epidemic when we wish to make a determination. Oftentimes, this difficulty can be mitigated by first choosing a set of candidate functions and finding the functions that best match the available observations/samples. This is the approach we take, which is explained in detail in subsequent sections.

4 Parameter Estimation and Model Selection

Suppose that $\mathbb{K} \subset \mathbb{R}_+^{d_{\max}}$, i.e., a set of nonnegative sequences of length d_{\max} , to which we assume \mathcal{K}_D and \mathcal{K}_U belong. Our goal is to identify the “best” sequences \mathcal{K}_D^* and \mathcal{K}_U^* in \mathbb{K} and parameter p_U^* , which minimize the discrepancy between the available observations and those predicted by the model, which is measured by some performance measure. In theory, this problem can be formulated as an optimization problem. But, as mentioned earlier, because this optimization problem has $1 + 2d_{\max}$ optimization variables, unless d_{\max} is small, which may not be the case in general, it would require many samples to find good estimates.

To address this issue, we propose to consider a family of sequences that are parameterized by a few parameters denoted by ξ . We assume that ξ belongs to a parameter set Ξ . For fixed parameter set Ξ , define \mathbb{K}^Ξ ($\subsetneq \mathbb{K}$) to be set of parameterized sequences. Since we approximate the time-varying transmissibility of infected DNs and UNs using parameterized sequences in \mathbb{K}^Ξ , we only need to estimate the parameters of the best sequences as well as the parameter p_U , which requires fewer samples. To this end, we employ an optimization problem.

Clearly, the quality of the solution we find by solving an appropriate optimization problem will depend on the allowed set \mathbb{K}^Ξ . Thus, some domain knowledge may be necessary to identify a suitable set of parameterized sequences that can be used to approximate the time-varying transmissibility. In Sect. 5, we adopt a family of scaled gamma kernels for the case study of COVID-19.

- **Predicted time series for fixed parameters:** Suppose that $\mathcal{T} = (T(n) : n = 1, 2, \dots, n_{\max})$ is the available finite time series of length $n_{\max} \geq 1$, where $T(n)$ is the total numbers of reported infections up to and including time n . Given a feasible solution $\mathbf{x} := (\tilde{p}_U, \tilde{\xi}_D, \tilde{\xi}_U) \in [0, 1] \times \Xi^2 =: \mathbb{X}$, where $\tilde{\xi}_D$ and $\tilde{\xi}_U$ are the parameters of \mathcal{K}_D and \mathcal{K}_U , respectively, we construct the following two sequences, $\tilde{\mathcal{N}}_D^\mathbf{x}$ and $\tilde{\mathcal{I}}_D^\mathbf{x}$: let $n_1 = \min\{n \in \mathbb{N} \mid T(n) \geq \tau_{\min}\}$, where $\tau_{\min} > 0$ is a minimum infection number parameter we select. Without loss of generality, we assume $n_1 = 1$. We then set $q(n_1) = T(n_1)/(P \times \tilde{p}_U)$, $\tilde{\mathcal{N}}_D^\mathbf{x}(n_1) = T(n_1)$ and $\tilde{\mathcal{N}}_D^\mathbf{x}(n') = 0$ for all $n' = n_1 - d_{\max} + 1, \dots, n_1 - 1$. Finally, starting with $n = n_1 + 1$, we use Eq. (3) to recursively generate the *predicted*

numbers of new cases $\tilde{N}_D^{\mathbf{x}}(n)$ for $n \in \{n_1+1, n_1+2, \dots, n_{\max}\}$, with the initial condition $\tilde{N}_D^{\mathbf{x}}(n_1) = (\tilde{N}_D^{\mathbf{x}}(n')) : n' = n_1 - d_{\max} + 1, \dots, n_1\}$. The predicted time series of the total number of reported infections is then given by $\tilde{\mathcal{I}}_D^{\mathbf{x}} = (\tilde{I}_D^{\mathbf{x}}(n) = \sum_{k=n_1}^n \tilde{N}_D^{\mathbf{x}}(k) : n = n_1, n_1 + 1, \dots, n_{\max})$.

- **Objective function:** To find the “best” parameters \mathbf{x}^* , we use the mean squared error (MSE) as a metric to measure the distance between the given time series \mathcal{T} and a predicted times series $\tilde{\mathcal{I}}_D^{\mathbf{x}}$:

$$\mathcal{D}(\mathcal{T}, \tilde{\mathcal{I}}_D^{\mathbf{x}}) = \sum_{n=n_1}^{n_{\max}} (T(n) - \tilde{I}_D^{\mathbf{x}}(n))^2 \quad (4)$$

- 1) **Two-population model (TPM):** In order to find optimal parameters \mathbf{x}^* that best approximate the given time series \mathcal{T} with respect to the MSE in (4) with two populations – DNs and UNs – we solve the following constrained optimization problem:

Opt. prob. for parameter estimation with two populations

$$\text{minimize}_{\mathbf{x} \in \mathbb{X}} \quad \mathcal{D}(\mathcal{T}, \tilde{\mathcal{I}}_D^{\mathbf{x}}) \quad (5)$$

We denote by \mathbb{X}^* the optimal set, i.e., the set of optimal points of (5), and v^* is the optimal value, i.e., $v^* = \mathcal{D}(\mathcal{T}, \tilde{\mathcal{I}}_D^{\mathbf{x}^*})$, where $\mathbf{x}^* \in \mathbb{X}^*$ is an optimal point.

- 2) **Single-population model (SPM):** To answer our question of whether or not there is a significant number of undetected infections that go unreported, we also solve the following related optimization problem only with a single population of DNs. First, from (3), when there are no undetected cases, we approximate the epidemic dynamics only with DNs using

$$N_D(n) = \mathcal{N}_D(n-1) \circledast \mathcal{K}_D(1 - q(n-1)), \quad n \in \mathbb{N}, \quad (6)$$

where $\mathcal{K}_D \in \mathbb{K}^{\Xi}$, and obviously the parameters \tilde{p}_U and $\tilde{\xi}_U$ play no role in the epidemic dynamics, which is clear from (6). Define $\mathbb{X}_S := \{\mathbf{x}' \in \mathbb{X} \mid p'_U = 0\} \subsetneq \mathbb{X}$.

Opt. prob. for parameter estimation with a single population

$$\text{minimize}_{\mathbf{x} \in \mathbb{X}_S} \mathcal{D}(\mathcal{T}, \tilde{\mathcal{I}}_D^{\mathbf{x}}) \quad (7)$$

We denote the optimal set and the optimal value of (7) by \mathbb{X}_S^* and v_S^* , respectively. Because $\mathbb{X}_S \subsetneq \mathbb{X}$, we have $v^* \leq v_S^*$.

4.1 Model Selection

Determining whether or not there is a considerable fraction of UN infections based on the optimal values of (5) and (7), can be cast as a model selection

problem. Loosely speaking, if all optimal points \mathbf{x}^* in \mathbb{X}^* have $p_U^* \approx 0$, then it likely suggests that at most a small fraction of infections may be undetected or unreported. On the other hand, if p_U^* is not close to zero for some optimal points in \mathbb{X}^* , it is still unclear if we should conclude that there are a significant number of unreported cases. This is especially the case if $v^* \approx v_S^*$ and the TPM does not significantly improve the approximation error (measured by the MSE) compared to the SPM. This is because the minor improvement in MSE may be due to ‘overfitting’, which is a well-known issue in model selection [11].

To cope with this problem, we borrow an idea from model selection theory, specifically information criteria [11]: in order to avoid overfitting, an information criterion, e.g., Akaike information criterion or Bayesian information criterion, is often used in model selection to carry out a trade-off between the accuracy of a chosen model and the number of model parameters. Unfortunately, these information criteria are not appropriate for our problem.

Instead, following a similar intuition behind information criteria, we base our decision on the relative improvement in the MSE we get using the TPM over the SPM: define $\theta^+ := (v_S^* - v^*)/v_S^*$ to be the relative improvement in MSE. We conclude that a significant fraction of cases come from UNs if (a) $\theta^+ \geq \theta_{th}$ and (b) $p_U^* \geq p_{th}$, where $\theta_{th}, p_{th} \in (0, 1)$ are thresholds we select.

5 Case Study Using Real Data: COVID-19

We carried out a case study using publicly available data for COVID-19. One of surprising aspects of COVID-19 which made its suppression more difficult, especially at the onset, is that many of infected individuals showed no serious outward symptoms, allowing them to unwittingly spread the infection and making contact tracing less effective. The reported estimates of asymptomatic cases vary considerably due to several difficulties, and the current best CDC estimate as of August 2021 is approximately 30%. Moreover, according to the CDC estimate, infected asymptomatic individuals (AIs) are about 75% as infectious as symptomatic individuals (SIs), posing a serious challenge [1].

We provide some numerical results that we obtained using the data downloaded from a public website that keeps track of the numbers of COVID-19 cases worldwide [2]. In this case study, reported cases come from infected DNs, whereas UNs do not report when infected, due to lack of serious symptoms or testing. Note that, in our setting, UNs may include SIs with mild symptoms which did not require testing/reporting, while the DNs may include some reported AIs confirmed through testing. However, most asymptomatic cases are assumed unreported. Thus, the fraction of UNs estimated using our approach will likely be larger than the fraction of AIs, denoted by p_A , as it includes unreported SIs.

- **Family of Gamma Distributions** – In our study, we approximate the time-varying transmissibility functions of infected DNs and UNs using a family of discrete-time approximations of gamma kernels: for given parameters

$\xi = (\alpha, \beta, r) \in \mathbb{R}_{++}^3 (= \Xi)$, the scaled discrete-time gamma kernel \mathcal{K}^ξ is given by a sequence $(K^\xi(n) : n = 1, 2, \dots, d_{\max})$, where

$$K^\xi(n) = r \frac{\beta^\alpha}{\Gamma(\alpha)} n^{\alpha-1} e^{-\beta n}, \quad n \in \{1, 2, \dots, d_{\max}\}, \quad (8)$$

Γ is the gamma function, and r is a scaling constant, which in our problem captures the basic reproduction number. We denote the (discrete-time) transmissibility function of DNs (resp. UNs) of the form in (8) by \mathcal{K}_D (resp. \mathcal{K}_U).

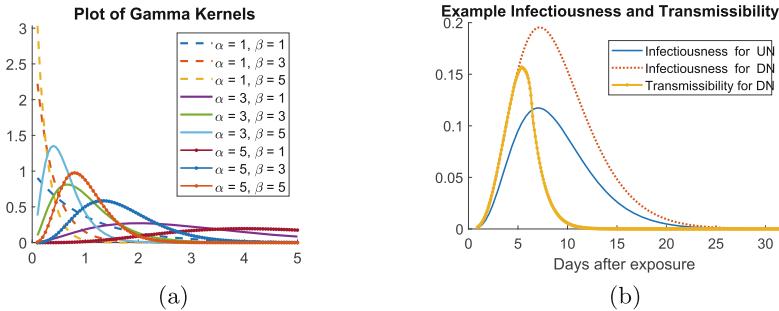


Fig. 1. (a) Family of gamma kernels for different parameter values, and (b) example infectiousness and transmissibility. In Fig. 1(b), we assume that the incubation period is 3–5 days and an infected DN isolates herself starting on day 5 or 6.

As shown in Fig. 1(a), gamma kernels with different values of parameters can be used to approximate functions with varying shapes and supports. Moreover, some of these gamma kernels closely resemble the example infectiousness functions reported in [8, Fig. 1], suggesting that gamma kernels are indeed good candidate functions for approximating the transmissibility functions. Since gamma kernels are probability density functions, their integrals are always equal to one. For this reason, we scale them to reflect the unknown average number of transmissions, i.e., reproduction numbers.

An exposed individual may become infectious as early as a few days into incubation period and remains contagious for several days after showing symptoms, followed by a slow decline in infectiousness [3]. However, once an infected DN shows symptoms and isolates herself, she may not transmit the disease much, indicated by low transmissibility. For an infected UN, on the other hand, her transmissibility will remain high, albeit slightly lower peak infectiousness than that of a DN as suggested in [1], for a longer period throughout her infectious period. These observations suggest that we should look for a gamma kernel with a wider support for the transmissibility function of UNs compared to that of DNs. Examples of infectiousness and transmissibility are shown in Fig. 1(b).

5.1 COVID-19 Outbreak Data Used in the Study

For our study, among many countries that experienced an outbreak, we selected four countries – France, Spain, the United Kingdom (U.K.), and the United States (U.S.) – and applied our approach using the publicly available daily numbers of reported cases in each country at the beginning of the pandemic [2].

Since the infectious period of an infected individual with moderate symptoms is believed to last about 7 to 12 days [4] and we wanted to consider the time series before widespread implementation of public health measures and policies in response, e.g., social distancing and face mask mandates, we considered the time series for a period of 30 days in early stages before the number of reported confirmed cases reached 10,000 cases in the countries.¹ The reported numbers we considered are for the period of 30 days from February 15, 2020 to March 15, 2020, except for the U.K., for which we considered the period of February 20, 2020 through March 20, 2020. The reason for this shift in the studied period for the U.K. is that there were 1,263 confirmed cases in the U.K. on March 15, 2020, which is far lower than the other three countries and our approach produced 500 estimates of p_U which were more spread out, indicating that there may not have been enough reported cases to produce consistent solutions.

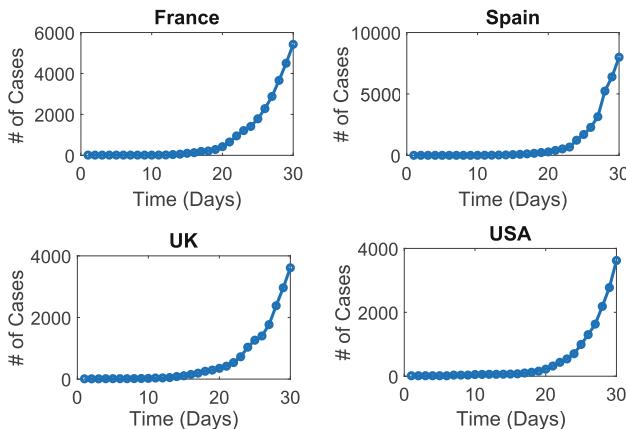


Fig. 2. Number of reported COVID-19 cases in France, Spain, the U.K., and the U.S.

The total number of reported cases over the periods under consideration for each country is shown in Fig. 2. It is clear that, despite some variations, the number of confirmed cases increases in a similar fashion. This is unsurprising because the number of infections is expected to grow exponentially at the onset of an epidemic before public health measures are introduced to curb the spread.

¹ Once widespread public health measures are in place, the spread dynamics are likely to change significantly with different parameters. For this reason, we considered the infection numbers in early stages.

5.2 Summary of Numerical Results

Before presenting the numerical results, we briefly describe a technical difficulty that arises due to the fact that the constrained optimization problems in (5) and (7) are nonconvex. In fact, our numerical studies suggest that there are many local optima, many of which are of poor quality. For this reason, we randomize the initial solution for parameter estimation, solve the optimization problems in equations (5) and (7) 500 times, and choose the best solution(s) out of the 500 solutions we produce as our best estimates of the optimal points.

We solved the optimization problems using the `fmincon` function built in MATLAB as well as `simulannealbnd` (simulated annealing) and `particleswarm` (particle swarm optimization).² Both `simulannealbnd` and `particleswarm` took much longer and produced solutions of comparable quality. Here we only report the solutions obtained using the `fmincon` function.

Table 1. Summary of numerical results.

	France	Spain	UK	USA
v^*	4.60×10^4	2.14×10^5	1.80×10^4	7.83×10^3
v_S^*	1.15×10^5	3.88×10^5	4.78×10^4	1.86×10^4
p_U^*	0.604	0.662	0.480	0.779
θ^+	0.60	0.45	0.62	0.58

Table 1 summarizes the numerical results, including the minimum MSEs achieved by the best solution out of the 500 solutions for the two models (SPM and TPM) as well as the corresponding θ^+ . It also provides the best estimate p_U^* of the fraction of UNs. It is clear from the reported numbers, especially θ^+ , that the TPM achieves a much smaller MSE than the SPM. Together with non-negligible estimates p_U^* , this suggest that there may be many more cases than reported, likely including a significant fraction of asymptomatic cases that went unreported in all four countries. This is consistent with several recent studies (e.g., [12]). In particular, the current CDC estimate [5] suggests that only 1 in 4.2 or roughly 23.8% of COVID-19 infections might have been reported.³ This is close to our estimate of 22.1% for the U.S.

Table 2. Five best estimates of p_U and MSEs for SPM and TPM (France).

Two-pop. model	p_U	0.6039	0.6114	0.4861	0.0899	0.2229
	MSE	4.60×10^4	4.81×10^4	5.85×10^4	7.29×10^4	9.21×10^4
Single-pop. model	MSE	1.15×10^5				

² Any mention of commercial products in this paper is for information only and is not intended for an endorsement or recommendation by NIST.

³ This estimate is for the period of February 2020 through May 2021.

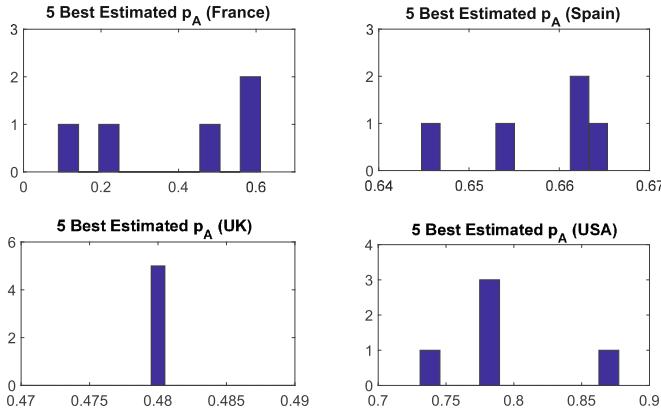


Fig. 3. Histogram of the five best estimated p_A (out of 500).

Figure 3 shows the distribution of the 5 best estimates of p_U (top 1%) for each country from the 500 solutions we obtained. To obtain these values, we first ordered the 500 solutions we generated using the TPM, by increasing value of MSE, and took the first 5 solutions with the 5 smallest MSEs.

As we can see from the figure, the 5 best estimates of p_U are concentrated around the best estimate for three countries – Spain, the U.K., and the U.S. In fact, the best 10 estimates for the U.K. are identical with the same MSE, suggesting that the obtained estimate $p_U^* = 0.48$ is likely a good estimate.

In contrast, for France only one other estimate is close to the best estimate, and two estimates are in fact considerably smaller. This is due to the difficulty explained at the beginning of this subsection that, in some cases, the optimization in (5) has many local optima and finding good solutions close to optimal point(s) is difficult. As a result, only a few solutions out of the 500 solutions we generated are good-quality solutions near optimal point(s); starting with the third best solution, the quality of the solution begins to deteriorate quickly with rapidly increasing MSE. This is illustrated in Table 2. On the other hand, the MSE achieved by the 5 best solutions of (7) for the SPM is identical. This is due to the fact that the parameter estimation for SPM is much easier than that for TPM; there are only 3 unknown parameters (α_S, β_S, r_S) to estimate for SPM compared to 7 unknown parameters in TPM.

As explained in Sect. 4, because our TPM treats *unreported* SIs as a part of UNs, the estimates p_U^* tend to overestimate p_A . For this reason, it is not our claim that p_U^* is an accurate estimate of p_A . Instead, large values of p_U^* together with significant relative improvements (θ^+) suggest that indeed there might be two different populations – DNs and UNs – that drove the epidemic dynamics at the onset of the pandemic, with the possibility of many asymptomatic cases.

Finally, we would like to mention that solving the two optimization problems in (5) and (7) 500 times for a country, using the MATLAB (version 2019b) `fmincon` optimization function takes less than 15 min on a laptop with an Intel

Core i7-2630QM CPU @ 2.0 GHz and 8 GB RAM. Hence, the proposed approach is computationally efficient.

6 Conclusion

We studied the problem of quickly determining whether or not there are many unreported cases of infection at the onset of an epidemic. Having an accurate answer to this question in early stages of an epidemic, will likely help the health experts devise better and more timely public health measures and policies. Our approach examines whether or not a TPM with both DNs and UNs can better predict the observed numbers than an SPM with only DNs. Our numerical studies using publicly available COVID-19 data in four countries suggest that a significant fraction of total infections in these countries may not have been reported, and the true numbers may be considerably larger.

References

1. <https://www.cdc.gov/coronavirus/2019-ncov/hcp/planning-scenarios.html>
2. <https://www.worldometers.info/coronavirus/>
3. <https://www.health.harvard.edu/diseases-and-conditions/if-youve-been-exposed-to-the-coronavirus>
4. <https://www.cdc.gov/coronavirus/2019-ncov/hcp/duration-isolation.html>
5. <https://www.cdc.gov/coronavirus/2019-ncov/cases-updates/burden.html>
6. Berger, N., Borgs, C., Chayes, J.T., Saberi, A.: On the spread of viruses on the internet. In: The Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA) (2005)
7. Bilge, L., Dumitras, T.: Before we knew it: an empirical study of zero-day attacks in the real world. In: The Proceedings of ACM Conference on Computer and Communications Security, pp. 833–844 (2012)
8. Fraser, C., Riley, S., Anderson, R.M., Ferguson, N.M.: Factors that make an infectious disease outbreak controllable. Proc. Natl. Acad. Sci. United States Am. (PNAS) **101**(16), 6146–6151 (2004)
9. Iribarren, J.L., Moro, E.: Branching dynamics of viral information spreading. Phys. Rev. E **84**, 046116 (2011)
10. La, R.J.: Cascading failures in interdependent systems: impact of degree variability and dependence. IEEE Trans. Netw. Sci. Eng. **5**(2), 127–140 (2018)
11. Linhart, H., Zucchini, W.: Model Selection, Wiley Series in Probability and Statistics. Wiley (1986)
12. Oran, D.P., Topol, E.J.: Prevalence of asymptomatic SARS-CoV-2 infection: a narrative review. Ann. Internal Med., M20–3012 (2020)
13. Subramanian, R., He, Q., Pascual, M.: Quantify asymptomatic infection and transmission of COVID-19 in New York City using observed cases, serology, and testing capacity. Proc. Natl. Acad. Sci. United States Am. **67**, 026126 (2021)



A Framework for Simulating Multiple Contagions Over Multiple Networks

Aparna Kishore, Lucas Machi, Chris J. Kuhlman^(✉), Dustin Machi,
and S. S. Ravi

University of Virginia, Charlottesville, VA 22904, USA
hugo3751@gmail.com, {ak8mj, lhm4v, cjk8gx, dm8qs, ssr6nh}@virginia.edu

Abstract. Many contagion processes evolving on populations do so simultaneously, interacting over time. Examples are co-evolution of human social processes and diseases, such as the uptake of mask wearing and disease spreading. Commensurately, multi-contagion agent-based simulations (ABSs) that represent populations as networks in order to capture interactions between pairs of nodes are becoming more popular. In this work, we present a new ABS system that simulates any number of contagions co-evolving on any number of networked populations. Individual (interacting) contagion models and individual networks are specified, and the system computes multi-contagion dynamics over time. This is a significant improvement over simulation frameworks that require union graphs to handle multiple networks, and/or additional code to orchestrate the computations of multiple contagions. We provide a formal model for the simulation system, an overview of the software, and case studies that illustrate applications of interacting contagions.

Keywords: Interacting contagions · Network discrete dynamical systems · Opinion dynamics · Multi-contagion agent-based simulation systems

1 Introduction

1.1 Background and Motivation

Many contagion processes evolving on populations do so simultaneously, interacting over time. Examples are co-evolution of social processes and diseases, such as increased hand washing during the influenza (flu) season and mask wearing during pandemic outbreaks, e.g., [9]; and co-transmission of information and evacuation decisions during natural disasters [28]. Commensurately, multi-contagion agent-based simulations (ABSs) that represent populations as networks in order to capture interactions between pairs of nodes (e.g., people are represented as nodes and interactions as edges in a network) are becoming more popular.

Currently, several simulation systems can compute interacting contagion dynamics, but they do so by incorporating data that must be preprocessed for

a particular simulation, or additional software must be written. For example, consider each of two interacting contagions, c_1 and c_2 , spreading on separate networks, G_1 and G_2 , respectively. Generally, there are common nodes and/or edges in the graphs so that there is interaction between contagions. Simulators are typically set up to store one graph, so that they require that a union graph G' be formed ($G' = G_1 \cup G_2$) where labels on edges designate whether an edge is in G_1 , G_2 , or both. Graph nodes are handled analogously. Consequently, if there are three graphs and simulations need to be performed with all combinations of two graphs, then three union graphs need be generated. With n_g of distinct graphs and n_{inst} graphs used in each simulation, the number of required union graphs is the binomial coefficient $C(n_g, n_{inst})$, which can be large.

Also, software must be added to frameworks to manage contagions. As an example for discrete time simulation, one might execute one contagion at each time step, so that if there are n_c contagions and t_{max} simulation time steps (e.g., days), a simulator might execute $n_c \times t_{max}$ time steps. But this approach adversely affects down-stream post-processing because “time” must be manipulated accordingly. For other frameworks, code has to be written to coordinate the execution of contagions at each time (see Sect. 1.3). Our system obviates the need for these changes.

We present a new ABS framework for modeling the spread of any number of contagions on any number of networks, using a structured and principled approach in both the software system and its use in applications. We also provide a formal model for our simulation system and case studies to illustrate its use in applications.

1.2 Our Contributions

1. Multi-Contagion Graph Dynamical Systems Formalism. We start with a graph dynamical systems (GDS) formalism [1] for single-contagion discrete dynamical systems. We extend that formalism to a *multiple contagion* GDS, MCGDS, operating over multiple graph instances. Any number $n_c \geq 1$ of contagions can operate on any number n_g of graphs, where $1 \leq n_g \leq n_c$, because while each contagion spreads on one graph, two or more contagions may spread on a single graph. The formal model is provided in Sect. 2.

2. Parallel Implementation of Multi-Contagion Simulation Framework. The software framework, called CSonNet (Contagion Simulation ON NETworks) is written in Python. Our simulation system implements concurrency through multi-processing. The system can simulate any number $n_c \geq 1$ of contagions operating over any number n_g of graphs, where $1 \leq n_g \leq n_c$. The graph instances, in general, can be disjoint, the same graph, or have common subsets of nodes and edges. Contagions can evolve independently or can interact as they evolve. Contagion models (e.g., SIR epidemic models, various threshold models for social contagions) are added to the system in one of two ways: (i) Python code that conforms to a specified interface is written and added to the system, or (ii) a user composes models through input files, using a set of

rules provided by the system. Option (i) is the only case where new code is written; all other inputs (e.g., specification of graphs, the mapping of contagions to the networks on which they evolve, rule-based models) are given through input files for a simulation. The system is overviewed in Sect. 3 and its strong scaling performance is evaluated in Sect. 4.

3. Case Studies. We provide two case studies, each with two non-interacting and two interacting contagion simulations: (i) SIR₁-SIR₂ epidemic contagions, and (ii) Threshold-SIR mixed social and epidemic contagions. Social networks with up to 75,877 nodes are used. While the system can compute contagions on much larger graphs, we choose graphs where fairly demanding multi-contagion simulations can be completed in less than 300 s of computational time, since in practice parametric simulation studies over large parameter spaces are typically required. These illustrate several features of CSonNet (see Sect. 5).

This simulation system significantly extends the single-contagion system described in [22]; that system does not implement the MCGDS of Contribution 1. Our implementation here (Contribution 2) is a large extension of the previous system in terms of capabilities (e.g., software added and software modified). None of the case studies here (Contribution 3) could be executed with the system in [22] without the workarounds specified in Sect. 1.1.

1.3 Related Work

Multiple Contagion Simulation Systems. To the best of our knowledge, no simulator has multi-contagion capabilities “out of the box” as is the case for our framework. However, because several frameworks enable models to be added to them (as does ours), it is possible to use these simulators for multiple-contagion scenarios, as described in Sect. 1.1. But these workarounds come with the price of requiring additional code and/or requiring more preprocessing of graphs before running simulations. Among the frameworks for simulations on networks that fit this description, across a range of capabilities, are: NetLogo [23], NDLIB [24], MASON [14], and Repast HPC [10].

Studies of Multiple Contagion Systems. Simulations of information spreading and evacuation decision-making in the context of hurricane evacuation modeling are discussed in [28]. Simulations of agents with limited attention spans for which multiple contagions (e.g., ideas) must compete are discussed in [27]. Agent-based models for competing languages are summarized in [20].

A number of papers have addressed models for competing and/or cooperating contagions (such as epidemics, product information, and misinformation in social media) over networks (see e.g., [16, 18, 27]). Polarization and consensus are studied with competing contagions [26]. One contagion that overtakes and halts a second contagion is studied in [7, 19]. A model for a simple contagion producing a complex contagion is given in [16]. Game theory is used to analyze

competing contagions in [11]. Many publications have studied optimization problems (e.g., seeding methods for influence maximization, minimizing the spread of contagions) in the context of multiple contagions (see e.g., [6, 7]).

2 Multi-contagion Graph Dynamical System

The underlying model implemented in our simulation system CSonNet is called a **graph dynamical system** (GDS). The formalism [1, 17] addresses a single contagion, but naturally extends to multiple contagions, which is done here. A GDS can simulate Turing machines for specific complexity classes [1, 4, 5].

2.1 Multiple Contagion GDS Formalism

A **multi-contagion GDS**, MCGDS, incorporating $n_c \geq 1$ contagions, is a quadruple $S(G^c, F^c, K^c, R^c)$: (i) G^c : a sequence of graphs $G^j(V^j, E^j)$, where G^j is the graph on which the contagion c_j propagates, V^j and E^j denote its vertex (i.e., node) and edge sets, $1 \leq j \leq n_c$; (ii) F^c : a sequence of sequences F^j , where each F^j is the sequence of local functions for contagion c_j ; (iii) K^c : a sequence of vertex (node) state sets K^j , where each K^j is the set of admissible vertex states for contagion c_j ; and (iv) R^c : a sequence of specifications R^j , where each R^j is the order in which local functions are executed for c_j . Each of these elements is defined below.

The graph $G^j(V^j, E^j)$ represents the **interaction network** for c_j . Let $n_j = |V^j|$ and $\mu_j = |E^j|$, with $n = |V^1 \cup V^2 \cup \dots \cup V^{n_c}|$ and $\mu = |E^1 \cup E^2 \cup \dots \cup E^{n_c}|$. Agents are vertices and pairwise agent interactions are edges in G^j . In general, edges may be directed or undirected. In this paper, for simplicity, we will consider graphs with undirected edges. Vertex (resp., edge) sets across contagions may be the same, disjoint, or have some common subset of nodes (resp., edges).

Each agent $v_i \in V^1 \cup \dots \cup V^{n_c}$ has an **agent state** or **vertex state** s_i which is a sequence $s_i = (s_{i,c_1}, s_{i,c_2}, \dots, s_{i,c_{n_c}})$ of n_c elements. Each $s_{i,c_j} \in K^j$. The **system state** (also called a **configuration**) $s = (s_1, s_2, \dots, s_n)$ is the sequence of n vertex states. Let the **local states**, denoted by $s[v_i]$, represent the sequence of states of vertex v_i and all of its distance-1 neighbors in each G^j . That is, $s[v_i] = (s_{c_1}[v_i], s_{c_2}[v_i], \dots, s_{c_{n_c}}[v_i])$, where $s_{c_j}[v_i]$ is the sequence of length $d_j(v_i) + 1$ of the states of v_i and each of its distance-1 neighbors in G^j for contagion c_j , and $d_j(v_i)$ is the degree of v_i in G^j . These quantities, at time t , are denoted by s_i^t , s_{i,c_j}^t , s^t , and $s^t[v_i]$ respectively.

Each agent v_i has a **local function** $f_{i,c_j} \in F^j$ that determines its state transitions for contagion c_j . The state of agent v_i at time $t + 1$ for c_j is given by

$$s_{i,c_j}^{t+1} = f_{i,c_j}(s^t[v_i]) \quad \text{for each } 1 \leq j \leq n_c. \quad (1)$$

Thus, the next state of v_i , with respect to each contagion c_j is a function of the current state of v_i and those of its distance-1 neighbors in *every* G^j over *all* contagions. That is, the argument on the right hand side of Equation (1) is

the same for each f_{i,c_j} , for a fixed v_i . This expression—and specifically the local functions for each contagion—indicates explicitly how each contagion affects (i.e., interacts with) every other contagion.

We assume that specification of **update order** R^j for the local functions for c_j corresponds to the **synchronous** update scheme. That is, all agents v_i evaluate their local functions f_{i,c_j} , $1 \leq i \leq n$, for c_j and update their states s_{i,c_j}^{t+1} *simultaneously* at each time step. Furthermore, we assume the updates across contagions are also done in parallel, i.e., all R^j are parallel with each other. Hence, f_{i,c_j} are computed in parallel for all $1 \leq i \leq n$ and for all $1 \leq j \leq n_c$ in Eq. (1). This enables greater parallelization of simulation computations, leading to more efficient calculations. However, other update orders R^j can be used. One example is a sequence W of the node IDs in G^j (of length n_j) and the local function for each node is computed in the order specified by W .

2.2 Example MCGDS

Two contagions propagate on the single network at the left in Fig. 1, where nodes are people and edges are interactions. The two contagions are: a social contagion c_1 on mask wearing and a disease contagion c_2 . Contagion c_1 uses a threshold model [12], with $K^1 = \mathbb{B} = \{0, 1\}$, where state 0 (resp., state 1) means that a person (node) is not (resp., is) wearing a mask. The local function f_{i,c_1} for all $i \in \{1, 2, 3, 4\}$ and c_1 is as follows. A node v_i changes from state 0 to 1 if at least a threshold $\theta = 1$ number of its neighbors are in state 1. Contagion c_2 is a disease model with states $K^2 = \{S, I, R\}$, where the states are susceptible (S), infectious (I), and recovered (R). The local function f_{i,c_2} for all $i \in \{1, 2, 3, 4\}$ and c_2 is as follows. If v_i is in state S, then v_i changes to state I with probability $p = p_{base} \cdot m_I \cdot m_S$ for each neighbor in state I, where $p_{base} = 0.008$, $m_I = 0.2$ if the infected neighbor is wearing a mask and $m_I = 1$ otherwise, and $m_S = 0.2$ if v_i is wearing a mask and $m_S = 1$ otherwise. If v_i is in state I and transitioned to state I at time t_I , then it transitions to state R at time $t = t_I + t_{inf}$, where $t_{inf} = 3$ is node v_i 's infectious duration. If v_i is in state R, it remains in state R. Contagion c_1 affects c_2 in that people that are wearing masks have a lesser probability of contracting and transmitting the disease.

The system states s at $t = 0, 1$, and 2 are given in Fig. 1. Contagion c_1 states are under “Social” and c_2 states are under “SIR.” At $t = 1$, c_1 spreads to each of v_2 and v_4 because these latter two nodes have v_1 as a neighbor, $s_{1,1}^0 = 1$, and the threshold for all nodes is $\theta = 1$. So at the end of $t = 1$, three of the four nodes are wearing masks. Also at $t = 1$, for contagion c_2 , v_3 is initially infected, but the Bernoulli trials for nodes v_2 and v_4 do not result in contagion spread, so the states remain $s_{2,2}^1 = s_{4,2}^1 = S$. (Note that at $t = 1$ for c_2 and node v_3 , $t < t_I + t_{inf} = 0 + 3$, so f_{3,c_2} returns the next state as $s_{3,c_2} = I$, which is the current state. The local function for v_3 will return the next state as I until $t = 3$, at which time $s_{3,c_2} = R$.) At $t = 2$ and for c_1 , v_3 changes to $s_{3,1}^2 = 1$ due to simple contagion spread from v_2 and v_4 . For c_2 at $t = 2$, the edge probability $p = p_{base} \cdot m_I \cdot m_S = (0.008)(1)(0.2) = 0.0016$ for the edge from v_3 to v_2 and from v_3 to v_4 because v_2 and v_4 are wearing masks at the end of $t = 1$ but v_3 is not.

The random number in $[0, 1]$ generated for the first edge is $0.0013 < p = 0.0016$ (i.e., the Bernoulli trial is successful) and so v_3 infects v_2 and $s_{2,2}^2 = I$. However, the Bernoulli trial for v_3 to infect v_4 is not successful because the drawn random number is $0.732 > p$, so $s_{4,2}^2 = S$ and therefore the states at $t = 2$ are as shown in Fig. 1. A case study using this two-contagion model is given in Sect. 5.1.

	time t=0			time t=1			time t=2		
	Node	Social	SIR	Node	Social	SIR	Node	Social	SIR
v_4				v_1	1	S	v_1	1	S
				v_2	0	S	v_2	1	S
v_1				v_3	0	I	v_3	0	I
				v_4	0	S	v_4	1	S

Fig. 1. Illustrative two-contagion dynamics for a MCGDS. Contagion c_1 is a social contagion of mask wearing represented by a threshold model; contagion c_2 is a disease contagion represented by an SIR model. Both contagions spread on the network on the left. Contagion c_1 affects c_2 . States for each contagion, for all nodes, are provided for three time steps.

3 CSonNet Modeling and Simulation Software System

3.1 Overview of Simulation Steps

CSonNet is a discrete-time multi-contagion ABS framework for networked populations. CSonNet implements the MCGDS model of Sect. 2 in the form of simulations. A **simulation** begins with reading in from file: (i) all graph instances G^c ; (ii) all local functions F^c over all nodes v_i , $1 \leq i \leq n$, and all contagions c_j , $1 \leq j \leq n_c$; (iii) the mapping $M_{j,\ell}$ of contagion c_j to graph G^ℓ ; (iv) the initial state assignments \mathbb{I} to all v_i for each c_j ; (v) the number n_i of iterations (i.e., simulation instances) to run; (vi) the maximum number t_{max} of time steps to run per iteration; and (vii) the number n_{wp} of worker processes that perform the computations. The number n_g of graphs and number n_c of contagions are determined from G^c and F^c , respectively. In particular, the number and types of contagions are completely specified by the local functions f_{i,c_j} of Eq. (1) in the sequence F^c and K^c . The local functions are not entered as equations in input files, but rather as models $M_{i,j}^{st}$, that implement these local functions for v_i and c_j . Section 3.2 below provides an example.

After reading all inputs, the main process of the simulation instantiates n_{wp} worker processes that carry out iterations in parallel on multi-core hardware computing nodes. The main process provides to each worker process the appropriate graphs, local functions, mapping of contagion to graph, initial conditions for particular iterations, and other parameters that the worker process needs, and then starts the worker processes. An **iteration** consists of computing the dynamics based on the states of all nodes at time $t = 0$ and over all contagions,

up through time t_{max} . Specifically, an iteration consists of computations over the following nested loops: (i) over all time steps $t \in [0, t_{max} - 1]$; (ii) for each time, over all contagions c_j , $j \in [1, n_c]$; and (iii) for each contagion, over all nodes v_i , $1 \leq i \leq n$. For each combination of (t, c_j, v_i) , Eq. (1) is evaluated, generating v_i 's next state s_{i,c_j}^{t+1} . If the next state is different from the current state of v_i , then this next state is written to file, along with the corresponding iteration number, t , c_j , and v_i ; this is the simulation output.

3.2 Agent State Transition Models from Rules

State transition models $M_{i,j}^{st}$, which are specified in input files, represent the local functions of Sect. 2. Below are examples of two state transition model files; each specifies the contagion number, followed by a row of entities that constitute the elements of a rule; all subsequent lines contain particular rule names and parameter values for these rules. Hence, a model is composed of rules. Currently, there is a fixed but extensible set of rules. Moreover, the model files below are used to perform a simulation such as that in Sect. 2.2.

Contagion 1 (c_1) has one rule. It is a (deterministic) threshold model that applies to all nodes, and describes the transition from state 0 to state 1 when at least 3 neighbors of a node are in state 1 (the cause). The threshold change based on influence from the SIR model is zero, meaning that the SIR contagion does not affect the threshold-based contagion. The last value of 1 represents a minimum threshold for the nodes, in the event of a threshold decrease; the threshold cannot go below 1.

Contagion 2 (c_2) is an SIR model. There are two rules, one for the state transition $S \rightarrow I$ and one for the transition $I \rightarrow R$. The transition $S \rightarrow I$ is governed by an edge probability of 0.006. If the infected (resp., susceptible) node is in state 1 for c_1 , then probability is reduced by the factor 0.5 (resp., 0.5). The Python list “[1,I]” indicates that states 1 from c_1 and state I from c_2 influence the transition $S \rightarrow I$. Thus, c_1 influences c_2 . The second rule states that a node stays in the infected state for 10 time units before transitioning $I \rightarrow R$.

```
Contagion 1
node from_state to_state cause rule param_1 param_2 param_3
all 0 1 [1] deterministic_progressive_node_threshold 0 3 1

Contagion 2
node from_state to_state cause rule param_1 param_2 param_3
all S I [1,I] edge_probability 0.006 0.5 0.5
all I R auto discrete_time_auto 10
```

4 Performance Evaluation

The networks in Sect. 4.1 are used to perform strong scaling studies of simulation time in Sect. 4.2.

4.1 Networks

Networks used in performance analyses and the case studies of Sect. 5 are given in Table 1. Two are face-to-face human social contact networks and the third (Epinions) is an online social network.

Table 1. The city-based human contact networks (first two entries) were made with the procedures in [3]. Each network is the giant component of the network, since we run dynamics on these networks. Property computations were performed with [2].

Network	Type	Num. nodes	Num. edges	Ave. deg.	Max. deg.	Ave. clus. coef.	Diameter
Danville, VA	Human contact	12961	44393	6.85	93	0.277	16
Newport News, VA	Human contact	64425	418879	13.00	344	0.261	22
Epinions	Online social	75877	405739	10.69	3044	0.138	15

4.2 Strong Scaling Results

Figure 2 shows strong scaling results for two-contagion simulations for each of the three networks. This is the total execution time over all worker processes (from the start of the first worker process to the end of the last executing process). The two contagions do not interact in Fig. 2a; there is interaction between contagions in Fig. 2b. The times are greater in the latter plot because all neighbors for both contagions must be iterated over to update the state of each contagion; these neighborhood iterations take place at each time step of each iteration for one simulation. Each data point, with \pm one standard deviation error bars, represents the results of ten simulations for each set of conditions. The data in both plots indicate that CSonNet exhibits strong scaling for independent and interacting contagions. The simulation conditions are appreciably onerous. Each time data point is for a simulation with 100 iterations, each over 100 time steps. In practice (e.g., for epidemic simulations), less than 100 iterations are performed (often on the order of 30), and typically about 14–30 time steps (days) are simulated. Yet, computations by worker processes complete in under 300 s for $n_{wp} = 32$ in Fig. 2b. We are interested in determining the sizes of networks on which simulations can be run in under five minutes because typical studies involve large parametric studies with many simulations.

5 Case Studies

Two multi-contagion case studies are presented. The first case study is a Threshold-SIR system spreading on the Newport News network, and is motivated in part by [8, 25]. The second is an SIR₁-SIR₂ system spreading on the Danville network. It is inspired by [15, 21]. In the first case study, one contagion inhibits the other; in the second study, two contagions reinforce each other. The purpose of these case studies is to demonstrate multi-contagion capabilities of the code.

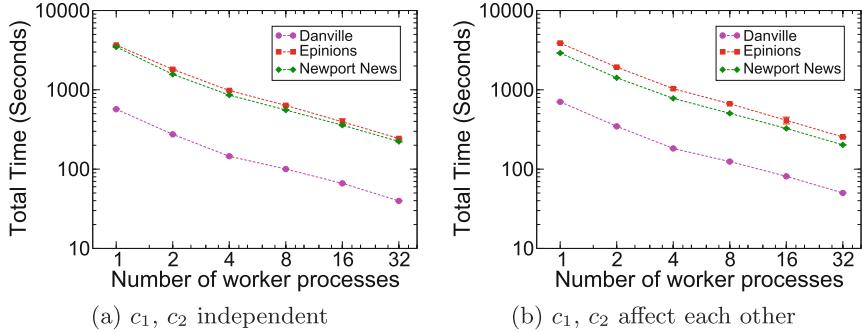


Fig. 2. Strong scaling of total execution time of worker processes. These strong scaling plots were generated for ten replicate simulations, where each simulation is run for 100 iterations. (a) Timing data for two non-interacting contagions, with one curve for each network. (b) Timing data are for interacting contagions.

5.1 Threshold-SIR Two-Contagion Model and Simulations

One contagion model (for c_1) is a Granovetter threshold model [12], with threshold $\theta = 3$, and is used to model mask wearing during COVID. States 0 and 1 indicate that a person is *not* wearing a mask and *is* wearing a mask respectively. A node transitions from state 0 to state 1 if at least θ of its neighbors are in state 1. The contagion model for c_2 is an SIR model for COVID. Contagion c_1 may affect c_2 as follows. Consider a person v_j in state I (i.e., infected with COVID) and another person who is v_i in state S (i.e., is susceptible). Each of v_i and v_j may or may not be wearing a mask during an encounter. When a person wears a mask, the probability of transmission is reduced by a multiplicative factor attributed to each person. For each person v_i wearing a mask, we let $m_i = 0.5$; otherwise, the factor is $m_i = 1$. Thus, the edge probability $p_{e,i}$ for transmission along the edge e between a susceptible person v_i and an infected person v_j is given by $p_{e,i} = w_{e,\text{base},i} \cdot m_i \cdot m_j$. In our case study, $w_{e,\text{base},i} = 0.006$ and the infectious duration $t_{\text{inf}} = 10$ days. COVID does not affect mask wearing in our model. The state transition model input files for this case study—for the interacting contagions simulation—were provided in Sect. 3.2.

Figure 3 provides cumulative infection curves (for c_2) and cumulative numbers of nodes in state 1 (for c_1) for the Newport News network. Seed nodes for each contagion are chosen uniformly at random and separately for each contagion: 0.0062 fraction of nodes for c_1 and 0.0062 fraction of nodes for c_2 in each iteration. The particular seed nodes are different for each of the 100 iterations, but the same seeds are used across the two simulations, for comparison. Figure 3a provides baseline data for non-interacting contagions. Figure 3b shows results for the case of mask wearing (c_1) affecting the spread of COVID (c_2). The latter plot indicates that mask wearing reduces the spread of COVID [13].

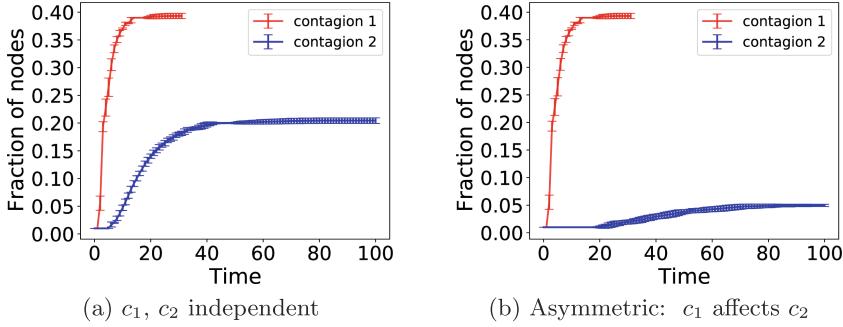


Fig. 3. Results from a two-contagion model on the Newport News network; c_1 is a threshold model for mask wearing and c_2 is an SIR model. There are two types of interactions: (a) independent contagions c_1 and c_2 (i.e., they do not affect each other) and (b) asymmetric contagions (i.e., contagion c_1 affects contagion c_2 , but not vice versa). Each data point in time is the average result over 100 iterations with error bars for \pm one standard deviation.

5.2 SIR₁-SIR₂ Two-Contagion Model and Simulations

In this SIR₁-SIR₂ system, if contagion c_1 affects contagion c_2 , then this is realized by increasing the probability that a node v_ℓ contracts c_2 , given that it has already contracted c_1 . More formally, consider a two-contagion system where the contagion models are SIR _{i} and SIR _{j} with $i, j \in \{1, 2\}$ and $i \neq j$. If the simultaneously evolving contagions do not interact, then $m_k = 1$ for contagion SIR _{k} ($k = 1, 2$) by definition. However, if contagion SIR _{i} affects SIR _{j} , and a node v_ℓ has already contracted contagion c_i , then the edge weight (i.e., probability of infection) for c_j , is $w_{e,j} = w_{e,base,j} \cdot m_j$. If $0 \leq m_j < 1$, then contracting contagion c_i reduces the probability of contracting c_j . If $m_j > 1$, then contracting contagion c_i increases the probability of contracting c_j . For SIR₁, the base edge weight is $w_{e,base,1} = 0.01$, the infectious duration is $t_{inf,1} = 10$ days, and for interacting contagions $m_1 = 8$. For SIR₂, $w_{e,base,2} = 0.005$, $t_{inf,2} = 12$ days, and for interacting contagions $m_2 = 5$. Since $m_1, m_2 > 1$, the contagions, when interacting, reinforce each other.

Figure 4 shows results from three simulations. Seed nodes for each contagion are chosen uniformly at random: 0.0077 fraction of nodes for c_1 and 0.0077 fraction of nodes for c_2 in each iteration. The particular seed nodes are different for each iteration, but the same seeds are used across the three simulations, for comparison. All plots are cumulative fractions of infected individuals as a function of time. Error bars at each time are \pm one standard deviations over the 100 iterations per simulation. In Fig. 4a, the two contagions do not interact. In Fig. 4b, c_1 affects c_2 , but not vice versa; so the cumulative fraction of infections increases for c_2 while remaining unchanged for c_1 . In Fig. 4c, the two contagions affect each other and the cumulative infected fractions increase for both contagions.

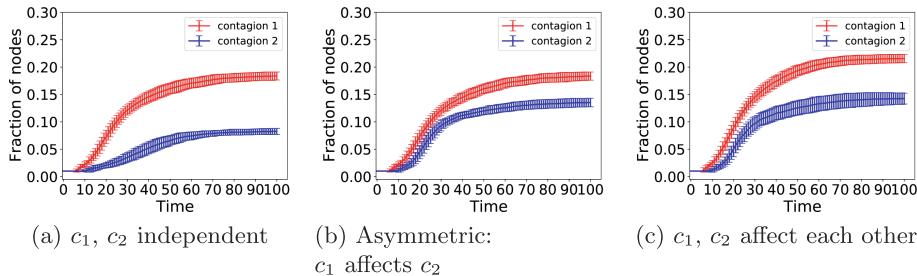


Fig. 4. Results from a two-contagion model on the Danville network where c_1 and c_2 are both SIR models. There are three types of interactions: (a) independent contagions c_1 and c_2 (i.e., they do not affect each other); (b) asymmetric contagions (i.e., contagion c_1 affects contagion c_2); and (c) symmetric contagions (i.e., c_1 and c_2 affect each other).

Acknowledgments. We thank the anonymous reviewers for their helpful feedback. We thank our colleagues at NSSAC and Research Computing at the University of Virginia for providing computational resources and technical support. This work has been partially supported by University of Virginia Strategic Investment Fund award number SIF160, NSF Grant OAC-1916805 (CINES), NSF Grant CMMI-1916670 (CRISP 2.0) and CCF-1918656 (Expeditions).

References

1. Adiga, A., et al.: Graphical dynamical systems and their applications to bio-social systems. *Int. J. Adv. Eng. Sci. Appl. Math.* **11**, 153–171 (2019)
2. Ahmed, N.K., Alo, R.A., Amelink, C.T., et al.: net.science: a cyberinfrastructure for sustained innovation in network science and engineering. In: *Gateway Conference*, pp. 71–74 (2020)
3. Barrett, C.L., et al.: Generation and analysis of large synthetic social contact networks. In: *Winter Simulation Conference (WSC)*, pp. 1003–1014 (2009)
4. Barrett, C.L., et al.: Complexity of reachability problems for finite discrete dynamical systems. *J. Comput. Syst. Sci.* **72**(8), 1317–1345 (2006)
5. Barrett, C.L., et al.: Modeling and analyzing social network dynamics using stochastic discrete graphical dynamical systems. *TCS* **412**(30), 3932–3946 (2011)
6. Borodin, A., Filmus, Y., Oren, J.: Threshold models for competitive influence in social networks. In: *International Workshop on Internet and Network Economics (WINE)*, pp. 539–550 (2010)
7. Budak, C., Agrawal, D., Abbadi, A.E.: Limiting the spread of misinformation in social networks. In: *WWW*, pp. 665–674 (2011)
8. Catching, A., Capponi, S., Yeh, M.T., et al.: Examining the interplay between face mask usage, asymptomatic transmission, and social distancing on the spread of covid-19. *Sci. Rep.* **11**, 1–11 (2021)
9. Cheng, V.C.C., Wong, S.C., et al.: The role of community-wide wearing of face mask for control of coronavirus disease 2019 (covid-19) epidemic due to sars-cov-2. *J. Infect.* **81**, 107–114 (2020)

10. Collier, N., North, M.: Parallel agent-based simulation with repast for high performance computing. *Simulation* **89**(10), 1215–1235 (2012)
11. Goyal, S., Kearns, M.: Competitive contagion in networks. In: Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing, pp. 759–774 (2012)
12. Granovetter, M.: Threshold models of collective behavior. *Am. J. Sociol.* **83**(6), 1420–1443 (1978)
13. Li, T., Liu, Y., Li, M., Qian, X., Dai, S.Y.: Mask or no mask for covid-19: a public health and market study. *PloS one* **15**(8), e0237691 (2020)
14. Luke, S., Balan, G.C., Sullivan, K., Panait, L.: MASON agent-based modeling framework (2019). <https://github.com/eclab/mason>, <https://cs.gmu.edu/~eclab/projects/mason/>
15. Martcheva, M., Pilyugin, S.S.: The role of coinfection in multidisease dynamics. *SIAM J. Appl. Math.* **66**(3), 843–872 (2006)
16. Min, B., Miguel, M.S.: Competing contagion processes: complex contagion triggered by simple contagion. *Sci. Rep.* **8**(10422), 1–8 (2018)
17. Mortveit, H.S., Reidys, C.M.: An Introduction to Sequential Dynamical Systems. Universitext, Springer, Heidelberg (2007). <https://doi.org/10.1007/978-0-387-49879-9>
18. Myers, S.A., Leskovec, J.: Clash of the contagions: cooperation and competition in information diffusion. In: 12th International Conference on Data Mining (ICDM), pp. 539–548 (2012)
19. Nguyen, N.P., Yan, G., Thai, M.T.: Analysis of misinformation containment in online social networks. *Comput. Netw.* **57**(10), 2133–2146 (2013)
20. Patriarca, M., Castello, X., Uriarte, J.R., Eguiluz, V.M., Miguel, M.S.: Influence of community structure on misinformation containment in online social networks. *Adv. Complex Syst.* **15**, 1250048-1–1250048-24 (2012)
21. Pawlowski, A., Jansson, M., Sköld, M., Rottenberg, M.E., Källenius, G.: Tuberculosis and HIV co-infection. *PLoS Pathogens* **8**(2), e1002464 (2012)
22. Priest, J.D., Kishore, A., et al.: Csonnet: an agent-based modeling software system for discrete time simulation. In: WSC (2021). <https://tinyurl.com/cnypt3u3>
23. Railsback, S., Ayllón, D., Berger, U., Grimm, V., Lytinen, S., Sheppard, C., Thiele, J.: Improving execution speed of models implemented in netlogo. *J. Artif. Soc. Social Simul.* **20**(1), 1–15 (2017)
24. Rossetti, G., Milli, L., et al.: Ndlib: a python library to model and analyze diffusion processes over complex networks. *Int. J. Data Sci. Anal.* **5**(1), 61–79 (2018)
25. Sahneh, F.D., Scoglio, C.: Epidemic spread in human networks. In: 50th IEEE Conference on Decision and Control and European Control Conference, pp. 3008–3013 (2011)
26. Vasconcelos, V.V., Levin, S.A., Pinheiro, F.L.: Consensus and polarization in competing complex contagion processes. *J. R. Soc. Interface* **16**, 20190196-1–20190196-8 (2019)
27. Weng, L., Flaminini, A., Vespignani, A., Menczer, F.: Competition among memes in a world with limited attention. *Sci. Rep.* **2**(335), 1–9 (2012)
28. Yang, Y., Mao, L., Metcalf, S.S.: Diffusion of hurricane evacuation behavior through a home-workplace social network: a spatially explicit agent-based simulation model. *Comput. Environ. Urban Syst.* **74**, 13–22 (2019)



Finding Influential Nodes in Complex Networks Using Nearest Neighborhood Trust Value

Koduru Hajarathaiah^(✉), Murali Krishna Enduri^(✉), and Satish Anamalamudi

Department of Computer Science and Engineering, SRM University-AP,
Amaravati, India

{hazarathaiah_koduru,muralikrishna.e,satish.a}@srmap.edu.in

Abstract. Information spreading in complex networks is an emerging topic in many applications such as social leaders, rumour control, viral marketing, and opinion monitor. Finding the influential nodes plays a pivotal role for information spreading in complex network. This is because influential nodes have capable to spread more information in compared with other nodes. Currently, there are many centrality measures proposed to identify the influential nodes in the complex network such as degree, betweenness, closeness, semi-local centralities and page-rank etc. These centrality measures are defined based on the local and/or global information of nodes in the network. Sheng et al. [18] propose centrality measure based on the information between nodes and structure of network. Inspired by this measure, we propose the nearest neighborhood trust page rank (NTPR) based on structural information of neighbours and nearest neighbours. We proposed the measure based on the similarity between nodes, degree ratio, trust value of neighbours and nearest neighbours. We also perform on various real world network with proposed centrality measure for finding the influential nodes. Furthermore, we also compare the results with existing basic centrality measures.

Keywords: Trust value · Influential nodes · Complex network · Centrality measure

1 Introduction

Analyzing and implementing the dynamics of the information spread in social, biological, technological networks become one of the most hot topic in the complex networks [15, 20]. In this context, we started to investigate the control of the dynamics of spread by using highly influential nodes. Highly influential nodes are more likely to spread the information in most of the network. So, identifying the influential nodes in the complex networks is a crucial problem in this area. We can also have a better knowledge on the structure and characteristics of the network while finding the influential nodes in the network.

Many researchers address this issue by proposing the centrality measures such as degree [8], betweenness [7], closeness [11], semi-local [3]. Recently, page-rank [2] and leader-rank [12] measures are considered as the importance of node's neighbours and neighbour's quality. Later, Nomura *et al.* [14] proposed Hyperlink Induced Topic Search (HITS) algorithm which is concentrated on finding the web link structures. This HITS algorithm focused on relevant information for a particular web page and gives ranks.

Many algorithms are focused on network's topology [4, 6, 10, 13, 23] and very few algorithms proposed on attribute information or local information of the node or nearest neighborhood information. Some of the researchers proposed centrality measures based on local information such as degree, semi-local, clustering coefficient [1, 17] normalized local centrality [23], local neighbor contribution [4], and local centrality with coefficient [22].

The relevance of nodes in networks may be affected by their interactions. In a social networks, communication between two individuals are based on their similarities. Sheng *et al.* [18] proposed an algorithm trust-PageRank (TPR), which is concentrated on the network's structure and attribute information between nodes in network. The measure TPR is the combination of trust value and pagerank algorithms. Trust value of two connected nodes is based on degree and similarity ratio of these two nodes. The similarity ratio of two individuals is defined on common characteristics of two nodes. The intuition behind this is to measure the TRP which is similarity and degree ratio that plays a key role in the information spreading within the network. But they considered TPR measure with in the neighborhood only. So we would like explore this measure not only to neighborhood but also next neighborhood. Even our work can be extended to further neighborhood levels.

In this paper, we proposed on Nearest Neighborhood Trust Page Rank (NTPR) which is focused on neighborhood trust value and nearest neighbours of trust value. In our work, degree ratio is also considered along with next neighbours instead of taking direct links. By including more level of neighbours in degree ratio captures, our measure NTPR for every vertex is more accurate. We observe neighbours and also nearest neighbour's information of node playing crucial role in influence of a node. Here trust value contains the similarity ratio, degree ratio with neighbours and also nearest neighbour's information. We give improved version of TPR which is proposed NTPR measure to finding seed nodes in complex networks. We perform on various real world network of this proposed centrality measure for finding the influential nodes. Furthermore, we also compare the results with existing basic centrality measures which are discussed in coming sections.

1.1 Related Work

From the last decade, several centrality measures have been implemented towards finding the important nodes in large-scale networks. We are giving some of the basic centrality measures in this section. We concentrated on undirected and unweighted network $G = (V, E)$, where V, E are vertices, edges. The degree centrality (DC) [8] is represented as the number of linkages occurring upon a

node. In closeness centrality (CC) [11], one vertex to other vertices average distance is calculated within the network. The CC of node v can be represented as

$$CC(v) = \frac{1}{\sum_{u \in V} d(v, u)}$$

Where $d(v, u)$ refers the distance between nodes u and v . The betweenness centrality (BC) [7] is a method for locating nodes that act as a link between two parts of a graph. The BC of node w can be calculated

$$BC(w) = \sum_{u \neq v \neq w \in V} \frac{d_{uv}(w)}{d_{uv}}$$

where d_{uv} is the shortest path from the vertex u to v (or v to u), $d_{uv}(w)$ is the shortest path between vertices u and v passing through vertex w . The semi-local centrality [3] is calculated based on the level of neighbours. Let SC of node v can be calculated as

$$SC(v) = \sum_{u \in N_v} \sum_{w \in N_u} NN(w)$$

where N_v and N_u are set of neighbours to vertices v and u respectively, and $NN(w)$ is next nearest neighbours of vertex v . Trust-PageRank [18] measure is constructed by introducing the trust value into the pagerank algorithm for finding the seed nodes. When a node is having higher trust-value then it receives more information. Based on the trust value, information is propagated to the neighbour nodes in networks. Trust-PageRank [18] measure defined as follows:

$$TPR_v^t = \frac{1 - \alpha}{n} + \alpha \sum_{u \in N_v} T(v, u) TPR_u^{t-1} \quad (1)$$

where $T(u, v)$ represents the trust value of nodes u and v , n is the nodes count and N_v is the neighbours of node v and α represents jump probability.

2 Nearest Neighbourhood Trust Page Rank

In this section, we proposed a Nearest Neighborhood Trust Page Rank (NTPR) which is focus on nearest neighbours of trust value and this trust value is based on similarity ratio and degree ratio. Similarity ratio and degree ratio with next nearest neighbours information are key contributions for defining this centrality measure. In the recent work [18], trust-PageRank (TPR) algorithm designed on combination of pagerank and trust-value. In this work consider higher level neighborhood information is missing. We observed that including the nearest neighbourhood information in trust value which contains similarity and degree ratio help us to investigate information spread by finding the influential nodes in the network.

Similarity Ratio: Similarity ratio of a vertex u to an neighbor v is the measure of similarity between the u and v nodes divided by the addition of the similarity

Algorithm 1: Algorithm for finding $NTPR_t$ for all vertices of G

```

Input: Graph  $G = (V, E)$ 
Output:  $NTPR_t$  measure of all vertices of  $G$ 
1 begin
2    $V = nodelist, E = edgelist, n := |V|$ 
3   for every edge  $(u, v)$  in  $E$  do
4     if  $u = v$  then
5        $s(u, v) = 1$ 
6     else
7        $s(u, v) = 0.1$ 
8   while  $t < Iteration$  do
9     For every  $(u, v)$  in  $E$ , find the  $S(u, v)$  by using Eq. 3.
10    For every  $(u, v)$  in  $E$ , find the  $SR(u, v)$  by using Eq. 2.
11    For every  $(u, v)$  in  $E$ , find the  $DR(u, v)$  by using Eq. 4.
12    For every  $(u, v)$  in  $E$ , find the  $TV(u, v)$  by using Eq. 5 with the help of
         $DR(u, v), SR(u, v)$  and  $k =$ .
13   for every vertex  $v$  in  $V$  do
14      $NTPR_0(v) = 0.1$ 
15   while  $t < Iteration$  do
16     for every vertex  $v$  in  $V$  do
17       find  $NTPR_t(v)$  by using Eq. 6 with help of  $TV, NTPR_{t-1}$ , and
          $\alpha = 0.85$ 
18   return  $(NTPR_t)$  /* final measure for each vertex*/

```

between the neighboring vertex v and its neighbor vertices w . The similarity ratio can be measured as follows:

$$SR(u, v) = \frac{S(u, v)}{\sum_{w \in N_v} S(v, w)} \quad (2)$$

The SimRank [19] is used for measuring the similarity between two vertices. In SimRank algorithm, if any two objects are similar then both are related to each other with some common characteristics. The similarity between u_1 and v_1 can be calculated as:

$$S(u, v) = \begin{cases} \frac{1}{|N_u||N_v|} \sum_{u_1 \in N_u} \sum_{v_1 \in N_v} S(u_1, v_1), & \text{if } u \neq v. \\ 1, & \text{if } u = v. \end{cases} \quad (3)$$

Where N_u is a neighbor node of u , N_v is neighbor node of v , $S(u_1, v_1)$ is similarity of u_1 and v_1 .

Degree Ratio: Degree ratio of a vertex u and neighbor vertex v is the ratio of the degree of vertex u to the addition of the degree of neighbor vertices of v .

The degree ratio can be defined as:

$$DR(u, v) = \frac{\sum_{u_1 \in N_u} d_{u_1}}{\sum_{w \in N_v} d_w + \sum_{w \in N_v} \sum_{k \in N_w} d_k} \quad (4)$$

Trust Value: Trust value of nodes u and v depends on both of similarity ratio and degree ratio of v to u . Trust value is calculated as follows:

$$TV(u, v) = (1 - k)SR(u, v) + kDR(u, v). \quad (5)$$

Where $SR(u, v)$ is similarity ratio of nodes u, v and $DR(u, v)$ is the degree ratio of nodes u and v . k is in between 0 and 1. We defining the Nearest Neighborhood Trust-PageRank based on the trust value which is as follows:

$$\begin{aligned} NTPR_t(v) = & \frac{1 - \alpha}{|V|} + \frac{\alpha}{|V|^2} \left[\sum_{u \in N_v} TV(v, u) NTPR_{t-1}(u) \right. \\ & \left. + \sum_{u \in N_v} \sum_{w \in N_u} TV(w, u) NTPR_{t-1}(w) \right] \end{aligned} \quad (6)$$

where $TV(v, u)$ is trust value of nodes v and u , $|V|$ represents number of vertices, α indicates jump probability. For finding the NTPR for every vertex in the network given as an algorithm procedure is represented in Algorithm 1. To get good performance of the influential nodes in the network and to avoid many parameters in the experiments, we consider the values of k and α is 0.85. The number of iterations we consider is 1000.

3 Implementation

Networks Description: We considered four real networks [16] for performance evaluation to test our proposed algorithm NTPR. These four networks are email-univ, euroroad, powergrid and web-polblogs networks. The details of networks are summarized in Table 1.

Table 1. Topological characteristics of 4 networks.

Networks	Nodes	Edges	Max Degree	Avg. Degree	Avg. Cluster Coeff.
Email-univ	1133	5451	71	9	0.220
Euroroad	1174	1417	10	2	0.017
Powergrid	4941	6594	19	2	0.080
Web-polblogs	643	2280	139	12.8	0.232

Susceptible-Infected-Recovered Model: In this research, to evaluate the proposed algorithm's performance SIR simulation model [5, 21] is used. The SIR

model is formed based on three compartments such as Susceptible, Infected, and Recovered. The susceptible refers to “not yet infected from others”. The infected refers to infected from others in the network. Finally recovered tells that after some iterations recovery has been happened that indicates no possibility of infection again. To find the spreading capability, initially some nodes are infected, later in each iteration these infected nodes are infects their neighbour nodes with the certain probability rate. Clearly, the cumulative infected nodes steadily increases with time and finally reaches a steady state. The SIR model is best for estimating the various centralities in terms of performance in the complex networks.

Table 2. Correlation (τ) between NTPR with DC, BC, CC, SC, TPR centralities.

Networks	$\tau_{(NTPR,DC)}$	$\tau_{(NTPR,BC)}$	$\tau_{(NTPR,CC)}$	$\tau_{(NTPR,SC)}$	$\tau_{(NTPR,TPR)}$
Email-univ	0.92234	0.73439	0.6748	0.7255	0.95262
Euroroad	0.65214	0.42884	0.04526	0.23629	0.93592
Powergrid	0.7759	0.52004	0.10505	0.37561	0.92017
Web-polblogs	0.81994	0.66339	0.46925	0.50709	0.82827

Kendall’s Tau (τ): The kendall rank correlation coefficient [9] is used to measure the performance and compare with existing measures. When any centrality gets higher τ value then that indicates accurate result. The kendall’s τ is the relationship between top ranked list which are generated by various centralities and rank list obtained by the SIR model of our new centrality. For example, if any two variable ranks are same then that relation is perfect correlation and the value is one. In case, any two variables are having opposite ranks then that relation is called negative correlation. So the tau is in between positive one to negative one value. In three cases, we can define kendall’s tau. Let us consider any two random variables K and T . *Case 1:* (k_i, t_i) and (k_j, t_j) are considered as concordant if both $k_i > k_j$ and $t_i > t_j$ or $k_i < k_j$ and $t_i < t_j$. *Case 2:* (k_i, t_i) and (k_j, t_j) are considered as discordant if both $k_i > k_j$ and $t_i < t_j$ or $k_i < k_j$ and $t_i > t_j$. *Case 3:* If both pairs are equal then they are considered as neither concordant nor discordant. The kendall’s tau is calculated as: $\tau = \frac{C-D}{0.5(N(N-1))}$ where C, D are the number of concordant and discordant pairs, N is the number of objects.

4 Results

In the results section, we discuss about correlation between our centrality measure NTPR with various existing centralities. We concentrated on SIR model analysis, performance wise comparison between NTPR with DC, BC, CC, SC, and TPR centralities.

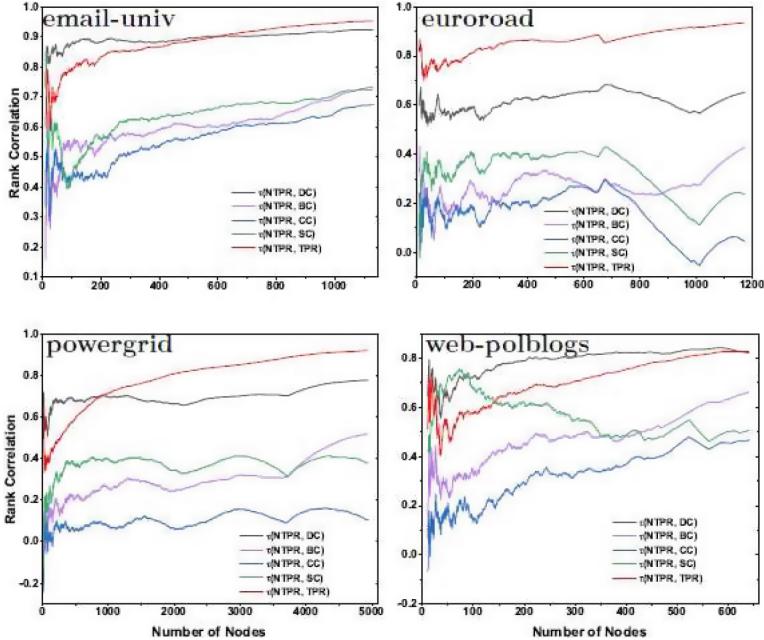


Fig. 1. Correlation (τ) between NTPR with different centralities such as DC, BC, CC, SC and TPR with top nodes.

4.1 Correlation of NTPR with Different Measures

In Table 2 and Fig. 1, we show the correlation between NTPR with other centralities such as DC, BC, CC, SC, and TPR centralities on four real world networks. We get the ranking for each node based on the centralities. For every top N vertices we can observe the correlation between NTPR and basic centrality measures which we plotted in Fig. 1 for every top N vertices where $N = 1, 2, \dots, n$ where n number of nodes in the network. If N is number of vertices of network (n) then correlation value of different centrality measures is shown in Table 2. We can observe that, how the NTPR is closely correlated with any basic centrality measure. From the Fig. 1 and Table 2, we clearly observe that NTPR is close to TPR only and it is far from any of the basic centralities. This is due to we have improved and defined based on TPR. From email-univ and powergrid network, degree is highly correlated with NTPR at initial set of nodes. Only in the web-polblogs, degree is highly correlated in later half of set of nodes. So NTPR is not good correlation with existing basic centralities except TPR.

4.2 Comparison Between Infection Rate and Centrality Value:

The NTPR centrality measure produced good results when compared with the DC, BC, CC, and TPR centralities. By using SIR model, initially single node

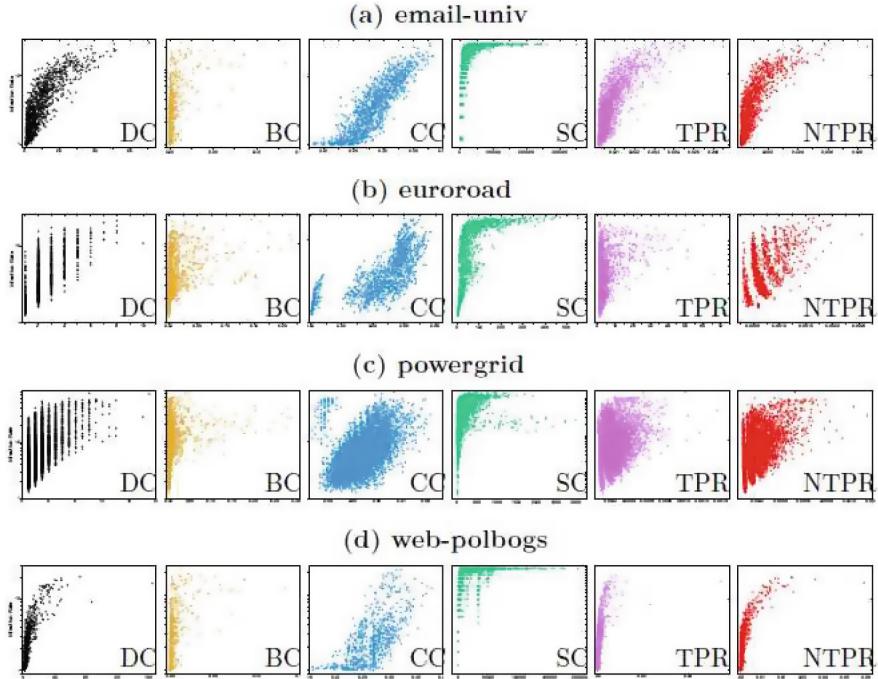


Fig. 2. In all four real world networks, comparison between infection rate and centrality value of SIR model.

accordingly their influenced values as infected node and find the cumulative infected after 100 simulations. The evaluation performance is simulated by SIR model which we show plots between the node influential values and infection rate in Fig. 2. In the email-univ network, our measure NTPR, TPR both are in performed well compare to DC, BC, CC which is shown in Fig. 2(a). The euroroad network in Fig. 2(b), our proposed measure NTPR is spreading information more than DC, BC, CC and TPR. The node influence is having high propagation capability with NTPR measure in the powergrid network. The BC measure is having poor performance when compared with basic centralities in powergrid network which can be observed from Fig. 2(c). In web-polblogs network, CC result is having spreading ability and also our centrality is also good. The web-polblogs SIR simulation results are represented in Fig. 2(d). The NTPR centrality measure performed consistently with accurate results in the powergrid and euroroad networks. In all four networks, our centrality measure NTPR generated with good results.

4.3 Comparison of Performance NTPR with Various Measures:

In this part, we show the results about the evaluation of infection spreading ability of top most influential nodes. The top ten influential nodes are discov-

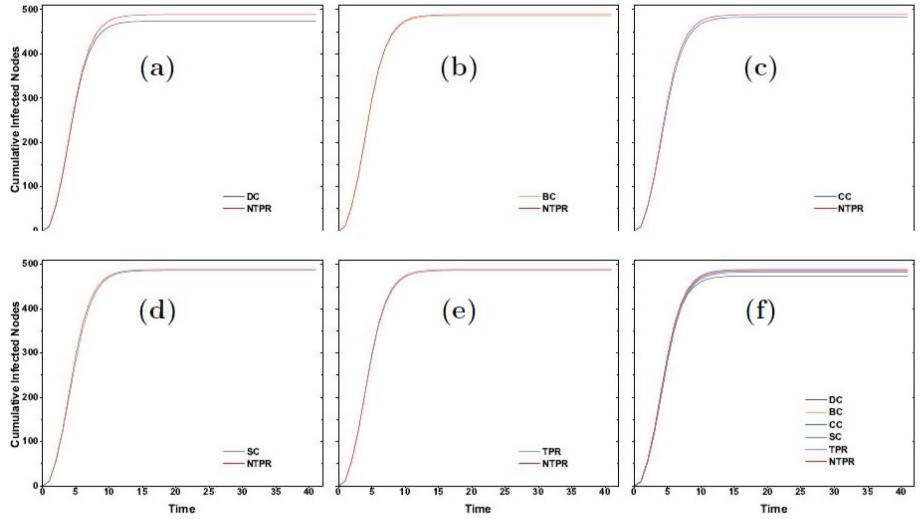


Fig. 3. The top-10 nodes are first infected nodes, as determined by the planned NTPR and other centralities using SIR model for the email-univ network.

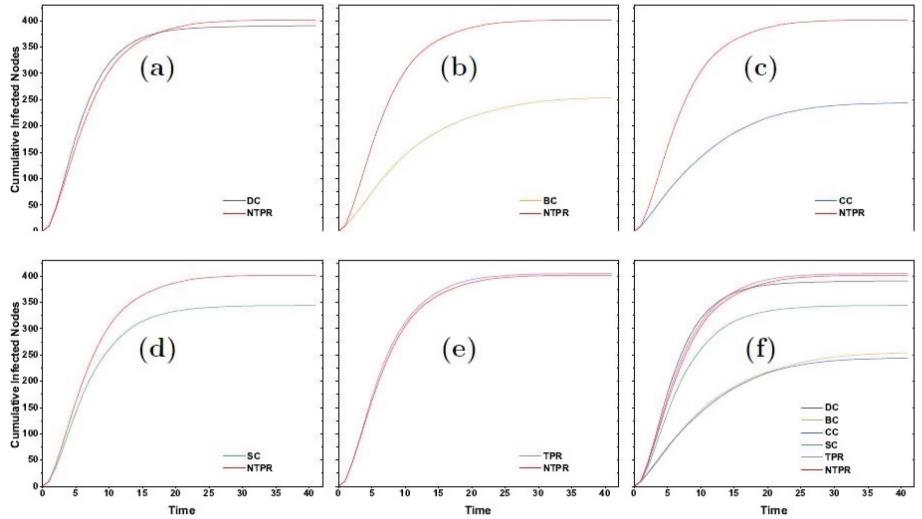


Fig. 4. The top-10 nodes are first infected nodes, as determined by the planned NTPR and other centralities using SIR model for the euroroad network.

ered by the different centralities like DC, BC, CC, SC, TPR, and NTPR. The SIR model [5] is used for the performance analysis with most important node's spreading ability. The infection rate beta β has taken 0.1 to 0.4. The recovery rate is considered as γ is one and 40 steps are propagated. The number of simulations has taken as 100. The NTPR centrality produced has better out-

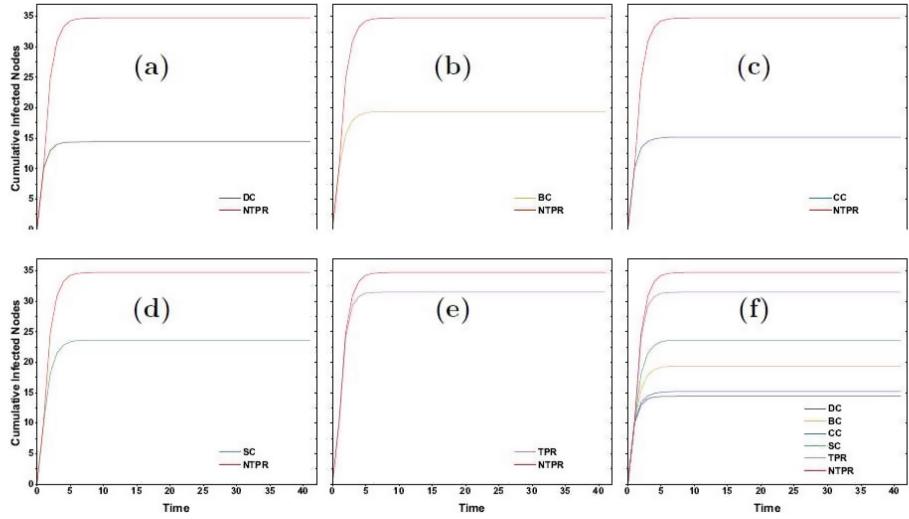


Fig. 5. The top-10 nodes are first infected nodes, as determined by the planned NTPR and other centralities using SIR model for the powergrid network.

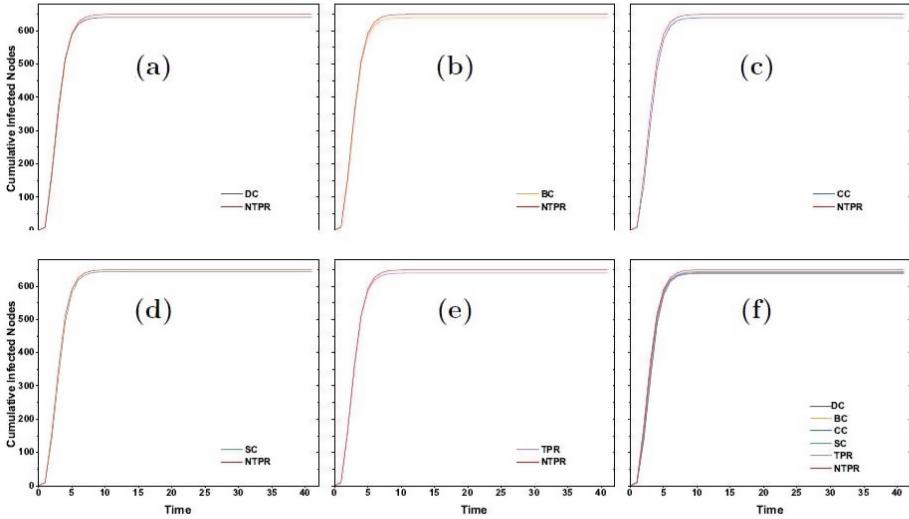


Fig. 6. The top-10 nodes are first infected nodes, as determined by the planned NTPR and other centralities using SIR model for the web-polblogs network.

come than DC, BC, CC, SC, and TPR. For the euroroad and powergrid real networks, our method gives outstanding results in comparison with the existing centralities. In addition, with web-polblogs network also our measure produce good outcome. The four network results are displayed in Figs. 3, 4, 5 and 6. In Fig. 3, cumulative infected node for NTPR measure is more, when compare with

other centrality measure. But BC, SC and TPR almost close with NTPR. In Fig. 4, most of the cases cumulative infected node for NTPR measure is dominating compare with other centrality measure. But DC and TPR almost close with NTPR. In Fig. 5, in all cases cumulative infected node for NTPR measure is outperforms compare with other basic centrality measure. In Fig. 6, cumulative infected node for NTPR measure is more close with other centrality measures.

5 Conclusions

In this work, authors concentrated on defining a centrality measure based on attribute information of nodes as well as network's topology. For the information of node's attribute, degree ratio is defined whereas for network's structure, the similarity ratio is used. In the existing work, the TPR is merged with pagerank and the trust-value. But in our work, we have concentrated on trust value with nearest neighborhood. The main intention of our measure is pagerank and trust-value along with the nearest neighbours consideration. The degree ratio also we go for the second level neighbours in network. By going up to second level, clearly we can analyze the dynamics of spread with in the network. The SIR model is used for evaluating the performance and the kendall's tau is used for finding closeness between NTPR and other existing basic centrality measures. We gave the experimental results by using our nearest neighborhood trust measure which the measure is used to find the influential nodes with in the network.

In the future work, instead of degree ratio we would like to consider the closeness, betweenness and some other centralities to analyze the outcomes with the proposed measure. Nearest neighborhood trust value will also have the possibility to extend more levels of neighbours (maximum number vertices of network or diameter of network).

References

1. Berahmand, K., Bouyer, A., Samadi, N.: A new centrality measure based on the negative and positive effects of clustering coefficient for identifying influential spreaders in complex networks. *Chaos Solitons Fractals* **110**, 41–54 (2018)
2. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. *Comput. Networks ISDN Syst.* **30**(1–7), 107–117 (1998)
3. Chen, D., Lü, L., Shang, M.S., Zhang, Y.C., Zhou, T.: Identifying influential nodes in complex networks. *Physica A* **391**(4), 1777–1787 (2012)
4. Dai, J., et al.: Identifying influential nodes in complex networks based on local neighbor contribution. *IEEE Access* **7**, 131719–131731 (2019)
5. Dorogovtsev, S.N., Goltsev, A.V., Mendes, J.F.: Critical phenomena in complex networks. *Rev. Mod. Phys.* **80**(4), 1275 (2008)
6. Erkol, S., Faqeeh, A., Radicchi, F.: Influence maximization in noisy networks. *EPL (Europhysics Letters)* **123**(5), 58007 (2018)
7. Freeman, L.C.: A set of measures of centrality based on betweenness. *Sociometry*, pp. 35–41 (1977)

8. Freeman, L.C.: Centrality in social networks conceptual clarification. *Soc. Networks* **1**(3), 215–239 (1978)
9. Kendall, M.G.: A new measure of rank correlation. *Biometrika* **30**(1/2), 81–93 (1938)
10. Kumar, S., Panda, B.: Identifying influential nodes in social networks: neighborhood coreness based voting approach. *Physica A* **553**, 124215 (2020)
11. Liu, H.L., Ma, C., Xiang, B.B., Tang, M., Zhang, H.F.: Identifying multiple influential spreaders based on generalized closeness centrality. *Physica A* **492**, 2237–2248 (2018)
12. Lü, L., Zhang, Y.C., Yeung, C.H., Zhou, T.: Leaders in social networks, the delicious case. *PLoS ONE* **6**(6), e21202 (2011)
13. Lv, Z., Zhao, N., Xiong, F., Chen, N.: A novel measure of identifying influential nodes in complex networks. *Physica A* **523**, 488–497 (2019)
14. Nomura, S., Oyama, S., Hayamizu, T., Ishida, T.: Analysis and improvement of hits algorithm for detecting web communities. *Syst. Comput. Japan* **35**(13), 32–42 (2004)
15. Opsahl, T.: The caenorhabditis elegans worm's neural network (2003). <https://toreopsahl.com/datasets/#celegans>
16. Rossi, R.A., Ahmed, N.K.: The network data repository with interactive graph analytics and visualization. In: AAAI (2015). <https://networkrepository.com>
17. Saramäki, J., Kivelä, M., Onnela, J.P., Kaski, K., Kertesz, J.: Generalizations of the clustering coefficient to weighted complex networks. *Phys. Rev. E* **75**(2), 027105 (2007)
18. Sheng, J., Zhu, J., Wang, Y., Wang, B., Hou, Z., et al.: Identifying influential nodes of complex networks based on trust-value. *Algorithms* **13**(11), 280 (2020)
19. Xuan, Z., Fengming, Z., Kewu, L., Xiao-Bin, H., Hu-Sheng, W.: Finding vital node by node importance evaluation matrix in complex networks. *Acta Physica Sinica* **61**(5), 1–7 (2012)
20. Yang, J., Yao, C., Ma, W., Chen, G.: A study of the spreading scheme for viral marketing based on a complex network model. *Physica A* **389**(4), 859–870 (2010)
21. Yang, R., Wang, B.H., Ren, J., Bai, W.J., Shi, Z.W., Wang, W.X., Zhou, T.: Epidemic spreading on heterogeneous networks with identical infectivity. *Phys. Lett. A* **364**(3–4), 189–193 (2007)
22. Zhao, X., Liu, F., Wang, J., Li, T., et al.: Evaluating influential nodes in social networks by local centrality with a coefficient. *ISPRS Int. J. Geo Inf.* **6**(2), 35 (2017)
23. Zhao, X., Xing, S., Wang, Q., et al.: Identifying influential spreaders in social networks via normalized local structure attributes. *IEEE Access* **6**, 66095–66104 (2018)



Two-Population SIR Model and Strategies to Reduce Mortality in Pandemics

Long Ma^(✉), Maksim Kitsak, and Piet Van Mieghem

Faculty of Electrical Engineering, Mathematics and Computer Science,
Delft University of Technology, PO Box 5031, 2600 Delft, The Netherlands
l.ma-2@tudelft.nl

Abstract. Despite many studies on the transmission mechanism of the Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), it remains still challenging to efficiently reduce mortality. In this work, we apply a two-population Susceptible-Infected-Removed (SIR) model to investigate the COVID-19 spreading when contacts between elderly and non-elderly individuals are reduced due to the high mortality risk of elderly people. We discover that the reduction of connections between two populations can delay the death curve but cannot reduce the final mortality. We propose a merged SIR model, which advises elderly individuals to interact less with their non-elderly connections at the initial stage but interact more with their non-elderly relationships later, to reduce mortality. Finally, immunizing elderly hub individuals can also significantly decrease mortality.

Keywords: COVID-19 pandemic · Mortality reduction · Two-population SIR model · Complex networks

1 Introduction

In many countries, the first wave of the Coronavirus disease 2019 (COVID-19) appeared in early 2020. In the summer of 2020, the spread of COVID-19 was significantly reduced due to strict restrictions [1] and weather effects [2]. At the beginning of autumn, students went back to school, marked the beginning of the second wave. However, rising infections were not taken seriously because infections were mainly among the young population and no significant hospitalization and deaths were observed [3]. Simultaneously, the high decease rate and self-preservation have caused that many elderly individuals reduced their contact with young people [4]. In October 2020, the hospitalization rates in many countries started increasing and the second COVID-19 wave was born. At the beginning of 2021, more contagious mutations of the coronavirus marked the third wave [5]. The third wave might not be the last COVID-19 pandemic and efficient strategies to reduce mortality will remain on the agenda.

The Susceptible-Infected-Removed (SIR) model [6,7] and its variations are commonly applied to describe the COVID-19 pandemic [8–10] and to forecast the number of infected and deceased cases in a population [11–13]. The ratio of new deceased elderly cases to new deceased non-elderly cases each day is expected to be constant over time in classic epidemic models but is time-varying in reality. Recent works start to consider the age-structured SIR model to describe the COVID-19 pandemic more realistic [14–18]. The age-structured SIR model divides the whole population into several age groups and the infection rates are age-dependent. Real data reveal that the elderly infected had a 30- to 100- fold higher risk of dying than younger individuals in many European countries [19]. Here, elderly and non-elderly individuals are respectively defined as individuals who are ≥ 65 years old and < 65 years old [19]. The elderly population accounts for a proportion of around 20% in many European countries [20]. Since the main difference in the COVID-19 pandemic is between elderly and non-elderly individuals, we construct a two-population SIR model [21] as follows:

1. There are two sub-populations: non-elderly and elderly individuals uniformly distributed over the social contact network. The virus spreading in a region is likely to start from non-elderly individuals because the virus can be carried into a community from other areas by commuters [22] and most commuters are non-elderly individuals.
2. There are four infection rates between and within non-elderly and elderly individuals. We believe that the highest infection rate is among elderly people. Elderly individuals are advised to a kind of self-isolation to protect themselves [23]. Staying in relative isolation from non-elderly people could be feasible, but some strong connections among elderly individuals, e.g., couples and people in the same nursing home, cannot be cut off. Conversely, the ties among elderly individuals will be stronger when their connections with non-elderly individuals are significantly reduced. The second highest infection rate is among the non-elderly population. The inter-group infection rates are the smallest since elderly individuals are afraid of being infected by non-elderly individuals. The infection rates between non-elderly and elderly individuals are low, but not zero, as elderly people still depend on younger people one way or another.

This article first investigates the features of fatality-related curves in the two-population SIR model when the connection between two populations is reduced. It shows that non-elderly deceased cases are prone to occur at the initial stage and most elderly deceased patients appear more often at a later stage. The final mortality, however, cannot be reduced by only limiting the connection between two populations. Moreover, reducing the infection rate among non-elderly individuals, e.g., closing schools, also cannot efficiently reduce mortality. In this work, we propose a merged SIR model to reduce the final mortality significantly. There are two stages in the merged SIR model: in the first stage, the model is the same as the two-population SIR model of Magal *et al.* [21] and in the second stage, the merged SIR model reduces to the standard SIR model. The physical meaning of the merged SIR model is that elder people are advised to reduce their connections with non-elderly individuals at the beginning of the pandemic and

interact more with non-elderly individuals later. The merged SIR model benefits the mortality reduction since many recovered non-elderly people can protect the susceptible elderly individuals.

Compartmental epidemic models assume that social contact networks are homogeneous with an infinite network size, but the actual network size is finite and the degree distributions of many real social networks follow a power law [24] with an exponent $\gamma \in [2, 3]$. We thus simulate the two-population SIR model on a scale-free network with a realistic network size to investigate the effect of network topology on the reduction of mortality. By comparing the simulation results of the two-population SIR model for the scale-free network and the Erdős-Rényi (ER) random network [25], the epidemic spreading in the heterogeneous network is much faster due to the star (or super spreader) effect. The reduction of connections between elderly and non-elderly individuals cannot decrease mortality in the compartmental epidemic model, but can reduce the mortality in the two-population SIR epidemic on the scale-free network. The merged SIR model is the best strategy to mitigate mortality. Finally, we illustrate that mortality can also be efficiently reduced by only immunizing rare elderly hub individuals.

2 Two-Population SIR Model

The two-population SIR model was first proposed by Magal *et al.* [21]. Similar models, that incorporate the underlying contact graph, are the networked SIR model proposed by Youssef and Scoglio [26], that was later entirely generalized to GEMF in [27]. Our work here applies the two-population SIR model to a realistic scenario related to the COVID-19 pandemic, systematically analyzes the death-related curve features, explores the effect of restrictions on mortality reduction and proposes an improved model to reduce the final mortality.

In the two-population SIR model, the non-elderly and elderly susceptible people are infected by non-elderly and elderly infectious individuals with different infection rates. Suppose that the elderly and non-elderly populations are well-mixed and large enough, the fractions of susceptible individuals $S(t)$, infectious individuals $I(t)$ and removed (recovered or deceased) individuals $R(t)$ at time t are reasonably well modeled by the following well-known differential equations:

$$\begin{cases} \frac{dS(t)}{dt} = -\text{diag}(S(t))BI(t), \\ \frac{dI(t)}{dt} = \text{diag}(S(t))BI(t) - EI(t), \\ \frac{dR(t)}{dt} = EI(t), \end{cases} \quad (1)$$

where $\text{diag}(*)$ denotes a 2×2 diagonal matrix whose diagonal entries are represented as a 2×1 vector. The vectors of fractions of susceptible individuals $S(t)$, infectious individuals $I(t)$ and removed individuals $R(t)$ for non-elderly and elderly individuals are respectively,

$$S(t) = \begin{pmatrix} S_n(t) \\ S_e(t) \end{pmatrix}, I(t) = \begin{pmatrix} I_n(t) \\ I_e(t) \end{pmatrix} \quad \text{and} \quad R(t) = \begin{pmatrix} R_n(t) \\ R_e(t) \end{pmatrix}. \quad (2)$$

The subscripts n and e denote non-elderly and elderly individuals, respectively. The matrices E (removed rates) and B (infection rates) are respectively

$$E = \begin{pmatrix} \delta_n & 0 \\ 0 & \delta_e \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} \beta_{nn} & \beta_{ne} \\ \beta_{en} & \beta_{ee} \end{pmatrix}, \quad (3)$$

where β_{ne} denotes the infection rate from elderly infectious individuals to non-elderly susceptible individuals, β_{en} denotes the infection rate from non-elderly infectious individuals to elderly susceptible individuals, β_{nn} denotes the infection rate among non-elderly individuals, β_{ee} denotes the infection rate among elderly individuals, δ_n denotes the removed rate for non-elderly infectious individuals and δ_e denotes the removed rate for elderly infectious individuals. To simplify, we let the infection rates between two populations be equal, $\beta_{ne} = \beta_{en} = \epsilon\beta_{nn}$, and thus the matrix B can be rewritten as

$$B = \beta_{nn} \begin{pmatrix} 1 & \epsilon \\ \epsilon & \kappa \end{pmatrix}. \quad (4)$$

We have that the non-elderly fractions $S_n(t) + I_n(t) + R_n(t) = N_n$ and the elderly fractions $S_e(t) + I_e(t) + R_e(t) = N_e$, where N_n denotes the fraction of non-elderly population and N_e denotes the fraction of elderly population. The two-population SIR model assumes that the total population is unchanged and thus $N_n + N_e = 1$. We denote the initial state by $v[0] = (S_n[0], I_n[0], R_n[0], S_e[0], I_e[0], R_e[0])$. A schematic depiction of the two-population SIR model is shown in Fig. 1. The infectious individuals will turn to be immune with a recovery rate (ξ_n for non-elderly individuals and ξ_e for elderly individuals) or deceased with a fatality rate (η_n for non-elderly individuals and η_e for elderly individuals). It holds that the removed rates $\delta_n = \eta_n + \xi_n$ and $\delta_e = \eta_e + \xi_e$. This work focuses on the fractions of new deceased non-elderly and elderly cases that are $\eta_n I_n(t)$ and $\eta_e I_e(t)$, respectively. We are also interested in the fractions of deceased non-elderly and elderly cases that are $D_n(t) = \eta_n R_n(t)/\delta_n$ and $D_e(t) = \eta_e R_e(t)/\delta_e$, respectively. By numerical solving Eqs. (1), we analyze the effect of infection rates on the following four death-related curve features,

1. maximum of $\eta_n I_n(t)$ and $\eta_e I_e(t)$: $\max_{t \geq 0} \eta_n I_n(t)$ and $\max_{t \geq 0} \eta_e I_e(t)$,
2. time points at which the maximum of $\eta_n I_n(t)$ and $\eta_e I_e(t)$ occur:
 $\underset{t \geq 0}{\operatorname{argmax}} \eta_n I_n(t)$ and $\underset{t \geq 0}{\operatorname{argmax}} \eta_e I_e(t)$,
3. time difference between two arguments of the maxima:
 $\underset{t \geq 0}{\operatorname{argmax}} \eta_e I_e(t) - \underset{t \geq 0}{\operatorname{argmax}} \eta_n I_n(t)$,
4. fractions of final deceased non-elderly cases $D_n(\infty)$ and elderly cases $D_e(\infty)$.

In this work, we set the fraction of non-elderly individuals as $N_n = 0.8$, the fraction of elderly individuals as $N_e = 0.2$ and the removed rates as $\delta_n = \delta_e = 0.1$. The fatality rates for non-elderly and elderly infections are set to be $\eta_n = 0.0001$ and $\eta_e = 0.01$, respectively. The initial state is set as $v[0] = (0.7999, 0.0001, 0, 0.2, 0, 0)$. These parameters are set based on real data. The elderly population makes up around 20% of the whole population in many

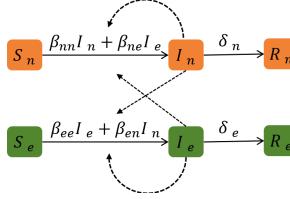


Fig. 1. Schematic depiction of the two-population SIR model. There are two populations: non-elderly people (highlighted in orange) and elderly people (highlighted in green). There are four different infection rates β_{nn} , β_{ne} , β_{en} and β_{ee} between and among the populations and two different removed rates δ_n and δ_e for each population.

European countries [20]. Elderly people who were infected had 30- to 100-fold higher risk of dying than younger people in several European countries [19]. The time to recovery or death is on average around 10 days [28]. We also investigate various parameter settings and find that the changing of these parameters has no much effect on the main conclusions drawn in this paper.

To understand the effect of parameters β_{nn} and κ in (4) on death-related curves, we plot the heatmaps as shown in Fig. 2. It indicates that there are large time difference $\underset{t \geq 0}{\text{argmax}} \eta_e I_e(t) - \underset{t \geq 0}{\text{argmax}} \eta_n I_n(t)$ when the infection rate β_{ee} is around the epidemic threshold δ_e/N_e (shown as the black curves in Fig. 2). This epidemic threshold is obtained assuming that the infection rate between two populations $\beta_{en} \rightarrow 0$. The mortality cannot be significantly reduced by only reducing the infection rate among non-elderly individuals β_{nn} , e.g., closing schools, given that the infection rate β_{ee} is above the epidemic threshold. The only efficient way to well reduce mortality is to keep the infection rate β_{ee} among elderly individuals below the epidemic threshold.

Although mortality can be well reduced by decreasing the infection rates among elderly individuals β_{ee} , this strategy is not realistic since elderly people need a sufficient amount of social interaction. This work discusses strategies to reduce mortality considering the social needs of all the people. Elderly people reduce their social connections with non-elderly individuals and increase their interactions with elderly relationships. Thus, their interaction frequency [29], which is the total number of social interactions per unit time, is unchanged. We study the effect of reducing connections between elderly and non-elderly individuals on mortality reduction by comparing the mortality in the standard SIR model and the two-population SIR model. To keep the interaction frequency in the standard SIR model and the two-population SIR model to be at the same level, the equivalent infection rate in the standard SIR model is

$$\beta = \beta_{nn} N_n^2 + \beta_{ee} N_e^2 + (\beta_{ne} + \beta_{en}) N_n N_e. \quad (5)$$

Figure 4a indicates that the fractions of the final deceased individuals for the standard SIR model and the two-population SIR model are the same. The effect

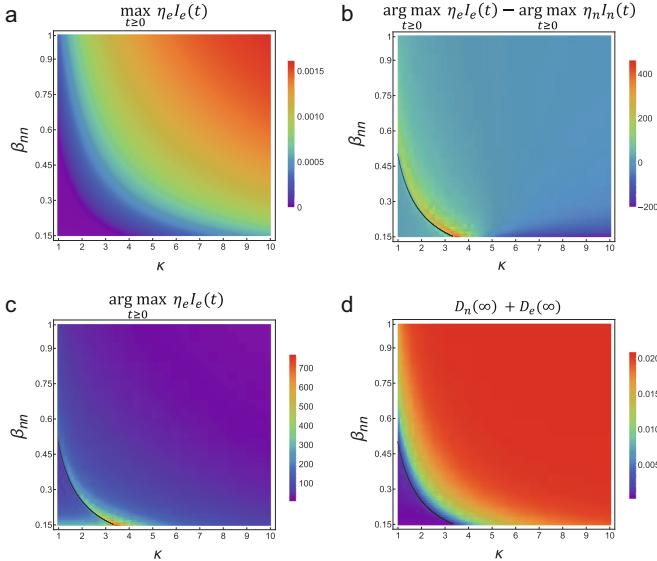


Fig. 2. Curve features for the two-population SIR model with different parameters β_{nn} and κ . The parameter ϵ is set to be $\epsilon = 0.001$. The black curves show the parameters when the infection rate β_{ee} is at the epidemic threshold δ_e/N_e . The time difference $\arg\max_{t \geq 0} \eta_e I_e(t) - \arg\max_{t \geq 0} \eta_n I_n(t)$ will be large when the infection rate β_{ee} is slightly above the epidemic threshold. The fraction of total deceased individuals will be small when the infection rate $\beta_{ee} < \delta_e/N_e$.

of reducing the connection between elderly and non-elderly groups is only to delay the deceased curve, but not to effectively reduce mortality.

3 Merged SIR Model to Reduce Mortality

To effectively reduce mortality, we propose a merged SIR model in which the epidemic spreading follows the two-population SIR model in the first stage and follows the standard SIR model in the second stage. The illustration of the merged SIR model is shown in Fig. 3. The reduction of the connection between two populations can delay the pandemic among elderly people. The reconnection of these two populations further protect elderly people due to the herd immunity effect of recovered non-elderly individuals. Figure 4 shows that the merged SIR model can significantly reduce the final deceased fractions and there is the best switch time point to minimize the final mortality. Heatmaps in Fig. 5 show the effect of parameters β_{nn} and ϵ on the best switch time point and reduced rate of the final mortality. The reduced rate of the final mortality is defined as

$$\frac{D_e(\infty) + D_n(\infty) - \tilde{D}_e(\infty) - \tilde{D}_n(\infty)}{D_e(\infty) + D_n(\infty)}, \quad (6)$$

where $D_e(\infty)$ and $D_n(\infty)$ are respectively the elderly and non-elderly mortality for the two-population SIR model and $\tilde{D}_e(\infty)$ and $\tilde{D}_n(\infty)$ are respectively the elderly and non-elderly mortality for the merged SIR model. Figure 5 reveals that the first stage (reducing the connection between two populations) should take a longer time if parameters β_{nn} and ϵ are smaller. Besides, the final mortality can be reduced more significantly for smaller parameters β_{nn} and ϵ .

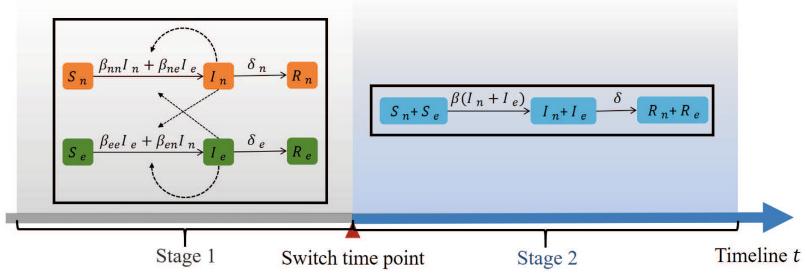


Fig. 3. Schematic depiction of the merged SIR model. This model has two stages: the first stage follows the two-population SIR model and the second stage follows the standard SIR model. The physical meaning of this model is to reduce the connection between the elderly and non-elderly populations initially and reconnect these two populations after many non-elderly infected individuals have been recovered. There is a switch time point between these two stages.

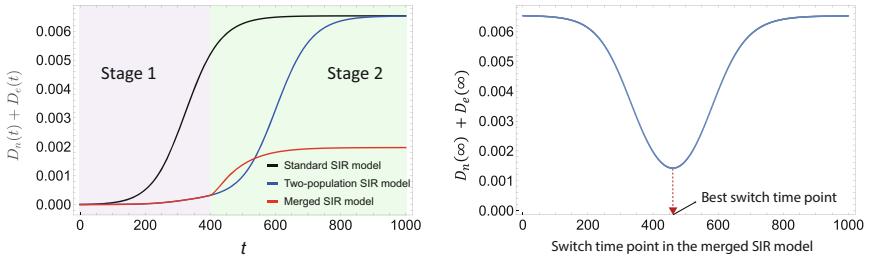


Fig. 4. The fractions of deceased individuals in the standard SIR model, the two-population SIR model and the merged SIR model. The parameters for the two-population SIR model are $\beta_{nn} = 0.15$, $\epsilon = 0.0001$ and $\kappa = 4$. We set the infection rate $\beta = 0.12$ for the standard SIR model to keep the interaction frequency the same among models. The left figure reveals that the two-population SIR model cannot, but the merged SIR model can efficiently reduce mortality. The right figure shows the final deceased fractions with different switch time points, indicating that there is the best switch time point to minimize the final mortality.

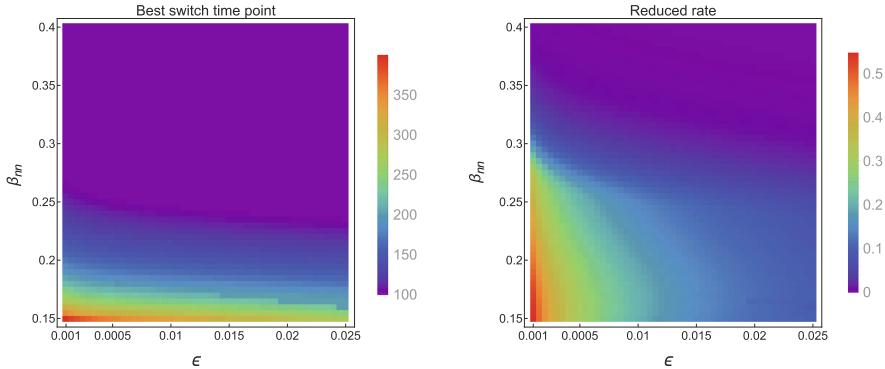


Fig. 5. The best switch time point and reduced rate of the final mortality. We choose different parameters β_{nn} and ϵ and a fixed parameter $\kappa = 4$ for the two-population SIR model. The best switch time point will be larger and the final mortality will be reduced more significantly if parameters β_{nn} and ϵ are smaller.

4 Two-Population SIR Epidemic on Large Complex Networks

We apply the Monte Carlo method [30] to simulate the two-group SIR epidemic on complex networks. In this work, we consider large networks with network size $N = 10^5$ generated by the configuration model [31] and the simulation starts from 100 non-elderly infected individuals. We first compare the simulation results on the scale-free network and the Erdős-Rényi random network to analyze the effect of network heterogeneity on epidemic curves. The network size N and mean degree $E[D]$ of the Erdős-Rényi random network are the same as the scale-free network. Figure 6a and Fig. 6b indicate that the epidemic spreading in the scale-free network is much faster than the spreading in the Erdős-Rényi random network due to the super spreaders. Figure 6c and Fig. 6d illustrate that the epidemic spreads quicker when the mean degree $E[D]$ is higher.

We simulate the standard SIR model, the two-population SIR model and the merged SIR model on the scale-free network as shown in Fig. 7. Different from the results as demonstrated in Fig. 4, for the epidemic spreading on complex networks, the final mortality for the two-population SIR model is lower than the standard SIR model since a part of susceptible elderly people can be protected by their recovered non-elderly relationships. This type of local immunity, which differs from the herd immunity, can only be observed in the epidemic spreading on networks. The merged SIR model is the best strategy to reduce mortality.

Given that there have been a limited number of COVID-19 vaccines, it is valuable to study the strategy to reduce mortality by immunizing specific population. There are rare elderly hub individuals in social networks, e.g., the priests, which are the virus's primary route of transmission from non-elderly to elderly people. Figure 8a and Fig. 8b reveal that the final mortality can be significantly reduced by only immunize 20 elderly hub individuals in 10^5 population assuming

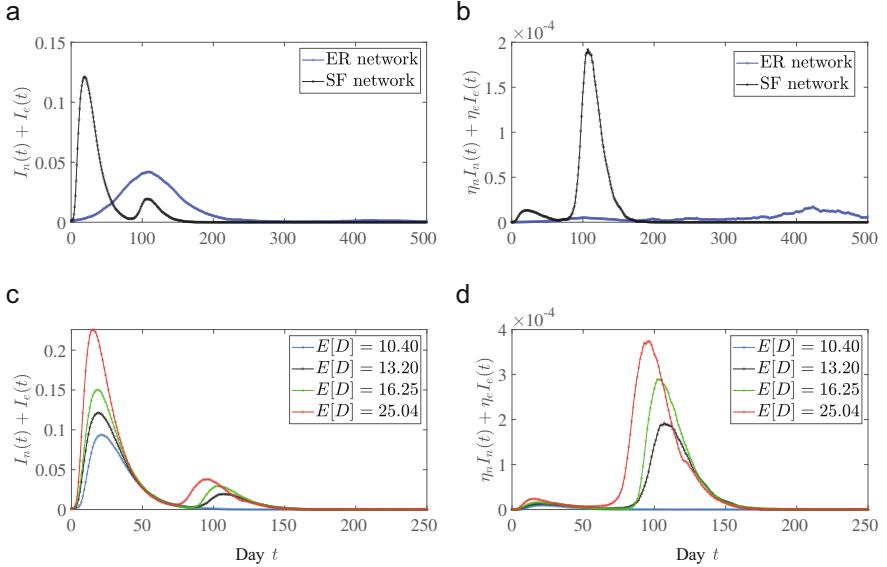


Fig. 6. Fractions of infectious cases $I_n(t) + I_e(t)$ and daily deceased cases $\eta_n I_n(t) + \eta_e I_e(t)$ for the two-population SIR epidemic on the scale-free network and the Erdős-Rényi network with the network size $N = 10^5$. The infection parameters are set to be $\beta_{nn} = 0.015$, $\kappa = 4$ and $\epsilon = 0.001$. The spreading in the scale-free network is much faster than the spreading in the Erdős-Rényi network. Figures c and d show the effect of mean degree $E[D]$ of the scale-free network on the two-population SIR epidemic. With the increase of the scale-free networks' link density, there are more individuals infected and deceased. The exponent in the scale free networks is set as $\gamma = 2.5$. The minimum degree of the scale-free network is set to be 5.

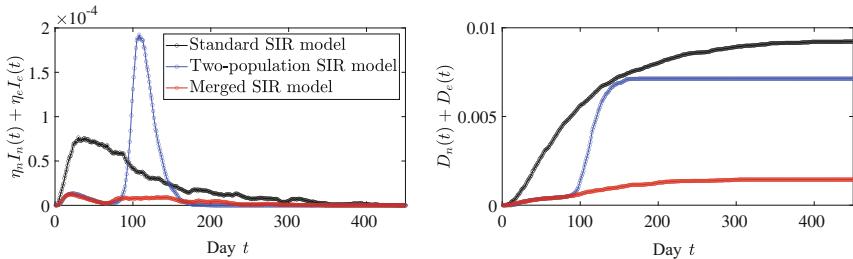


Fig. 7. Fractions of daily deceased cases $\eta_n I_n(t) + \eta_e I_e(t)$ and deceased cases $D_n(t) + D_e(t)$ for the standard SIR epidemic, the two-population SIR epidemic and the merged SIR epidemic on the scale-free network with the network size $N = 10^5$. The infection rate in the standard SIR model is $\beta = 0.012$. The infection rates in the two-population SIR model are the same as Fig. 6. Different from the result in Fig. 4, the final deceased fraction for the two-population SIR model is lower than the standard SIR model. The final deceased fraction for the merged SIR model is the smallest, indicating that the merged SIR model is the best strategy to reduce the mortality.

that the vaccines are 100% effective. We further analyze the situation when the vaccines are 80% effective. Figure 8c and Fig. 8d illustrate that more elderly hub individuals require to be immunized to reduce mortality efficiently.

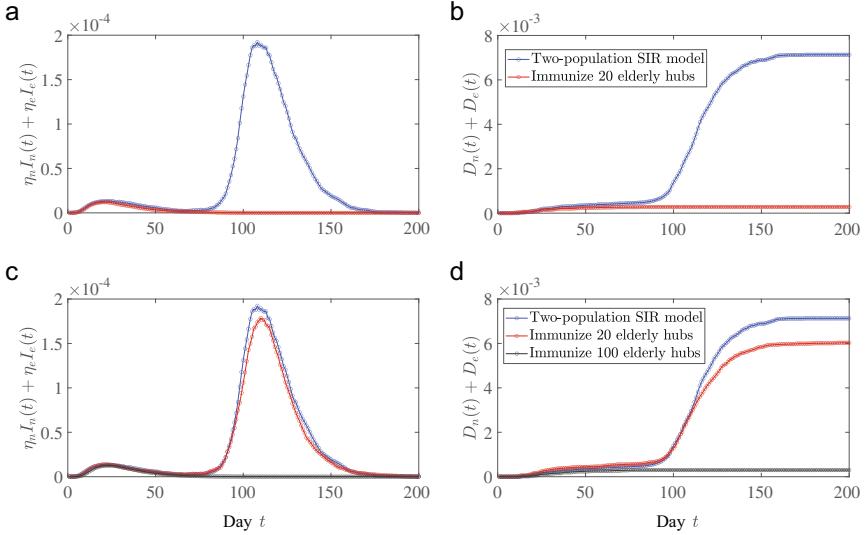


Fig. 8. Effect of immunizing rare elderly hub individuals on reducing the final mortality. Figures a and b respectively show the fractions of daily deceased cases $\eta_n I_n(t) + \eta_e I_e(t)$ and deceased cases $D_n(t) + D_e(t)$ for the two-population SIR epidemic with and without immunizing elderly hub individuals. We immunize 20 elderly individuals with the largest degree in the simulation of the two-population SIR model on the scale-free network with 10^5 population assuming that the vaccines are 100% effective. Figures c and d show the fractions when the vaccines are 80% effective. It requires more vaccine doses to effectively reduce the mortality if the vaccines are less effective.

5 Conclusions

Since early 2020, scientists have found that COVID-19 is substantially more dangerous for the elderly. Elderly people's interactions with their non-elderly relationships are reduced to lower the risk of being infected and deceased. This work applies the two-population SIR model to study if the mortality can be efficiently reduced when the connections between elderly and non-elderly individuals are significantly reduced. It reveals that severing ties between two populations can postpone the pandemic but not effectively cut mortality. We further propose the merged SIR model and find that reconnecting two populations at an appropriate time can significantly lessen the final mortality. Assuming that rare vaccines are available, this study recommends immunizing elderly hub individuals first to better decrease mortality. Future works may extend the merged SIR model

to more complicated models considering more realistic scenarios, e.g., the time-varying infection rates, the exposed state and the situation that some recovered individuals return to being susceptible.

References

1. Jordan, R.E., Adab, P., Cheng, K.K.: COVID-19: risk factors for severe disease and death (2020)
2. Merow, C., Urban, M.C.: Seasonality and uncertainty in global COVID-19 growth rates. *Proc. Natl. Acad. Sci.* **117**(44), 27456–27464 (2020)
3. Reimann, N.: More young people are dying of coronavirus in Florida, as state shatters death record tuesday (2020)
4. Armitage, R., Nellums, L.B.: COVID-19 and the consequences of isolating the elderly. *Lancet Public Health* **5**(5), e256 (2020)
5. Leung, K., Shum, M.H.H., Leung, G.M., Lam, T.T.Y., Wu, J.T.: Early transmissibility assessment of the N501Y mutant strains of SARS-CoV-2 in the United Kingdom. *Eurosurveillance* **26**(1), 2002106 (2021)
6. Fernández-Villaverde, J., Jones, C.I.: Estimating and simulating a SIRD model of COVID-19 for many countries, states, and cities. Technical report, National Bureau of Economic Research (2020)
7. Morris, D.H., Rossine, F.W., Plotkin, J.B., Levin, S.A.: Optimal, near-optimal, and robust epidemic control. *Commun. Phys.* **4**(1), 1–8 (2021)
8. Cooper, I., Mondal, A., Antonopoulos, C.G.: A SIR model assumption for the spread of COVID-19 in different communities. *Chaos Solitons Fractals* **139**, 110057 (2020)
9. Faranda, D., Alberti, T.: Modeling the second wave of COVID-19 infections in France and Italy via a stochastic SEIR model. *Chaos Interdisc. J. Nonlinear Sci.* **30**(11), 111101 (2020)
10. Ceylan, Z.: Estimation of COVID-19 prevalence in Italy, Spain, and France. *Sci. Total Environ.* **729**, 138817 (2020)
11. Prasse, B., Achterberg, M.A., Ma, L., Van Mieghem, P.: Network-inference-based prediction of the COVID-19 epidemic outbreak in the Chinese province Hubei. *Appl. Network Sci.* **5**(1), 1–11 (2020). <https://doi.org/10.1007/s41109-020-00274-2>
12. Massimo A. Achterberg, Bastian Prasse, Long Ma, Stojan Trajanovski, Maksim Kitsak, and Piet Van Mieghem. Comparing the accuracy of several network-based COVID-19 prediction algorithms. *International journal of forecasting*, 2020
13. Neves, A.G.M., Guerrero, G.: Predicting the evolution of the COVID-19 epidemic with the A-SIR model: lombardy, Italy and Sao Paulo state, Brazil. *Phys. D Non-linear Phenomena*, **413**, 132693 (2020)
14. Arenas, A., et al.: A mathematical model for the spatiotemporal epidemic spreading of COVID-19. *MedRxiv* (2020)
15. Rădulescu, A., Williams, C., Cavanagh, K.: Management strategies in a SEIR-type model of COVID-19 community spread. *Sci. Rep.* **10**(1), 1–16 (2020)
16. Di Domenico, L., Pullano, G., Sabbatini, C.E., Boëlle, P.-Y., Colizza, V.: Impact of lockdown on COVID-19 epidemic in île-de-france and possible exit strategies. *BMC Med.* **18**(1), 1–13 (2020)
17. Prem, K., et al.: The effect of control strategies to reduce social mixing on outcomes of the COVID-19 epidemic in Wuhan, China: a modelling study. *Lancet Public Health* **5**(5), e261–e270 (2020)

18. Zhao, Z.-Y., Zhu, Y.-Z., Jing-Wen, X., Shi-Xiong, H., Qing-Qing, H., Lei, Z., Rui, J., Liu, X.-C., Wang, Y., Yang, M., et al.: A five-compartment model of age-specific transmissibility of SARS-CoV-2. *Infect. Dis. Poverty* **9**(1), 1–15 (2020)
19. Ioannidis, J.P.A., Axfors, C., Contopoulos-Ioannidis, D.G.: Population-level COVID-19 mortality risk for non-elderly individuals overall and for non-elderly individuals without underlying diseases in pandemic epicenters. *Environ. Res.* **188**, 109890 (2020)
20. Worldbank, D.: Population ages 65 and above (% of total population) (2019)
21. Magal, P., Seydi, O., Webb, G.: Final size of an epidemic for a two-group SIR model. *SIAM J. Appl. Math.* **76**(5), 2042–2059 (2016)
22. Pei, S., Kandula, S., Yang, W., Shaman, J.: Forecasting the spatial transmission of influenza in the United States. *Proc. Natl. Acad. Sci.* **115**(11), 2752–2757 (2018)
23. Hajek, A., König, H.-H.: Social isolation and loneliness of older adults in times of the COVID-19 pandemic: can use of online social media sites and video chats assist in mitigating social isolation and loneliness? *Gerontology* **67**(1), 121–124 (2021)
24. Eubank, S., Anil Kumar, V.S., Marathe, M.V., Srinivasan, A., Wang, N.: Structural and algorithmic aspects of massive social networks. In: Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 718–727. Citeseer (2004)
25. Erdős, P., Rényi, A.: On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* **5**(1), 17–60 (1960)
26. Youssef, M., Scoglio, C.: An individual-based approach to SIR epidemics in contact networks. *J. Theor. Biol.* **283**(1), 136–144 (2011)
27. Sahneh, F.D., Scoglio, C., Van Mieghem, P.: Generalized epidemic mean-field model for spreading processes over multilayer complex networks. *IEEE/ACM Trans. Networking* **21**(5), 1609–1620 (2013)
28. Centers for Disease Control and Prevention. Clinical questions about COVID-19: questions and answers (2020)
29. Simmel, G.: Über sociale Differenzierung: sociologische und psychologische Untersuchungen, vol. 10. Duncker & Humblot (1890)
30. Juher, D., Ripoll, J., Saldaña, J.: Analysis and monte carlo simulations of a model for the spread of infectious diseases in heterogeneous metapopulations. *Phys. Rev. E* **80**(4), 041920 (2009)
31. Newman, M.E.J.: The structure and function of complex networks. *SIAM Rev.* **45**(2), 167–256 (2003)



Hypergraph Laplacians in Diffusion Framework

Mehmet Emin Aktas¹(✉) and Esra Akbas²

¹ Department of Mathematics and Statistics, University of Central Oklahoma,
Edmond, OK 73034, USA
maktas@uco.edu

² Department of Computer Science, Oklahoma State University,
Stillwater, OK 74078, USA
eakbas@okstate.edu

Abstract. Modeling diffusion on networks is an important concept in network science. It helps to understand how an idea, information, or infection, diffuses within the network. The graph Laplacian has been used to model diffusion on graphs for years. Extending graph Laplacians to hypergraphs is not an intuitive task since hyperedges can include more than two vertices, and edge incidence and vertex adjacency are set-valued in hypergraphs. To handle this issue, researchers limit their attention to specific hypergraphs, which is often not the case for real-world hypergraphs, or reduce hypergraphs to graphs, where these reductions result in information loss. In this paper, we present two new hypergraph Laplacians that can be defined on any hypergraphs. Our Laplacians take the relations between hyperedges into consideration, hence can be used to model diffusion on hypergraphs not only between vertices but also hyperedges. As an application, we study the Enron network and show the effectiveness of the proposed Laplacians in the influential node detection problem. These Laplacians can be further employed in different hypergraph mining problems, such as social contagion models on hypergraphs, influence study on hypergraphs, hypergraph classification, and hypergraph representation learning.

Keywords: Hypergraph · Laplacian · Simplicial complex · Diffusion

1 Introduction

Modeling diffusion on networks is an important concept in network science. It helps to understand how an idea, information, or infection, diffuses within networks based on network topology, the pattern of who is connected to whom. For example, in a social network, modeling information diffusion can be useful in rumor controlling [22], and in epidemic networks, diffusion models can help with controlling the spread of a disease [30]. The graph Laplacian has been used to model diffusion on networks for years [19, 20, 25, 27, 30].

On the other hand, as we see in different real-world applications, such as human communication, chemical reactions, and ecological systems, interactions can occur in groups of three or more nodes. They cannot be simply described as pairwise relations [3], rather should be described as *higher-order* interactions. Hence, the rich higher-order interactions are lost in the basic network model and we need to take higher-order interactions into consideration for a more accurate representation of complex systems. As one solution to this problem, *hypergraphs* are used to model complex systems [17, 31]. In a hypergraph, nodes again represent entities as it happens for graphs, but differently, a hypergraph has *hyperedges* for higher-order interactions in the network.

Extending graph Laplacians to hypergraphs is not an intuitive task since hyperedges can include more than two vertices, and edge incidence and vertex adjacency are set-valued in hypergraphs. To handle this issue, researchers limit their attention to uniform hypergraphs, where hyperedges have the same cardinality [10, 16]. But this is not realistic since real-world hypergraphs are almost never uniform. As another approach, researchers reduce non-uniform hypergraphs to graphs using line graph and clique expansion, but these reductions unsurprisingly result in information loss [11, 21]. Besides, some researchers study random walks on hypergraphs for constructing Laplacians, but many of them only consider uniform hypergraphs [9, 23, 24], and in the non-uniform case, these random walks are equivalent to a random walk on the graph clique expansion of the hypergraph [5–7, 14]. Furthermore, the authors in [18, 26] represent a hypergraph as a simplicial complex, i.e., subsets of hyperedges are also hyperedges, but this is often not the case in real-world hypergraphs.

As another solution, Horak et al. [15] defines simplicial Laplacians for hypergraphs, which can be considered as the more realistic one in diffusion framework since it allows to define diffusion between nodes and also hyperedges for non-uniform hypergraphs. However, it has three severe issues. First, simplicial Laplacians are only defined for the hypergraphs with a simplicial complex structure. Second, for a hyperedge of size k , the simplicial Laplacian models the diffusion only *through* the hyperedges of sizes $k - 1$ and/or $k + 1$. However, in the diffusion framework, information on a hyperedge can diffuse through other hyperedges regardless of their sizes. Third, when we use the simplicial Laplacians in modeling diffusion, we need to assume that information only diffuses *between* fixed size hyperedges. However, a hyperedge can affect other hyperedges regardless of their sizes.

To address the issues discussed above, we develop new and more general hypergraph Laplacians inspiring from the simplicial Laplacians. Our Laplacians are defined for any hypergraph, not necessarily with a simplicial complex structure. This addresses the first issue. The first hypergraph Laplacian we propose, \mathcal{L}_k , allows defining diffusion between hyperedges of a fixed-sized k *through* any hyperedges, which addresses the second issue. The second hypergraph Laplacian we propose, \mathcal{L}_H , allows defining diffusion between *any* hyperedges through any hyperedges, which addresses the third issue. We further employ the proposed Laplacians in modelling the epidemic spreading on hypergraphs, as an application.

The paper is formatted as follows. In Sect. 2, we first give the necessary preliminaries and background on graphs, hypergraphs, and Laplacians. In Sect. 3, we develop two new hypergraph Laplacians to address the needs. In Sect. 4, we study the influential node detection problem on the Enron network using the proposed Laplacians. Our final remarks with future work directions are found in Sect. 5.

2 Preliminaries and Background

In this section, we discuss the preliminary concepts for graphs, hypergraphs, graph Laplacian and hypergraph Laplacian. We also elaborate on related work with a particular focus on the hypergraph Laplacian that uses simplicial complex.

2.1 Graphs and Hypergraphs

Graphs are structured data representing relationships between objects [1, 8]. They are formed by a set of *vertices* (also called nodes) and a set of *edges* that are connections between pairs of vertices. In a formal definition, a network G is a pair of sets $G = (V, E)$ where V is the set of vertices and $E \subset V \times V$ is the set of edges of the network. If there is a score for the relationship between vertices that could represent the strength of interaction, we can represent this type of relationships or interactions by a *weighted network*. In a weighted network, a weight function $W : E \rightarrow \mathbb{R}$ is defined to assign a weight for each edge.

Let G be a weighted undirected graph with the vertex set V and a weight function $w : V \times V \rightarrow \mathbb{R}^{\geq 0}$. The *adjacency matrix* A of G is defined as the $n \times n$ matrix with $A(i, j) = w(v_i, v_j)$ for $i, j \in \{1, \dots, n\}$ with n being the number of vertices of G . Furthermore, let D be the $n \times n$ diagonal matrix with $D(i, i) = \sum_j w(i, j)$, i.e., the weighted degree of the vertex $i \leq n$. We can define the graph Laplacian L as $L = D - A$ where D is the weighted degree matrix and A is the weighted adjacency matrix. The graph Laplacian, first appeared in [20] where the author analyzed flows in electrical networks, is an operator on a real-valued function on vertices of a graph. The graph Laplacian has been thoroughly studied and applied in network science for years [25]. The spectrum of Laplacian is related to many graph features such as connected components, spanning trees, centralities, and diffusion [27].

A *hypergraph* H denoted by $H = (V, E = (e_i)_{i \in I})$ on the finite vertex set V is a family $(e_i)_{i \in I}$ (I is a finite set of indexes) of subsets of V called *hyperedges*. Hyperedges can be in different sizes possibly ranging from one vertex $\{v\} \subseteq V$ to the entire vertex set V .

2.2 Simplicial Laplacians

In this paper, we develop hypergraph Laplacians inspiring from the simplicial complex hypergraph representation, namely the *simplicial Laplacian* (or Hodge Laplacian). That is why, in this section, we explain how to construct the simplicial Laplacian.

We start with defining the simplicial complex. A *simplicial complex* is a topological object which is built as a union of points, edges, triangles, tetrahedron, and higher-dimensional polytopes, i.e. *simplices*. A 0-simplex is a point, a 1-simplex is two points connected with a line segment, a 2-simplex is a filled triangle, etc. (see Fig. 1).

More formally, a simplicial complex K is a finite collection of simplices such that every face of a simplex of K belongs to K and the intersection of any two simplices of K is a common face of both of them. In graphs, 0-simplices correspond to vertices, 1-simplices to edges, 2-simplices to triangles, and so on. We denote an i -simplex as $\sigma = [v_0, \dots, v_i]$ where $v_j \in V$ for all $j \in \{0, \dots, i\}$.

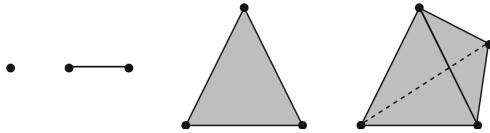


Fig. 1. 0-, 1-, 2-, and 3-simplex from left to right (borrowed from [2]).

We now start defining the simplicial Laplacian. Let $S_p(K)$ be the set of all p -simplices of a simplicial complex K . An i -chain of K over the field \mathbb{R} is a formal sum of its i -simplices and i -th *chain group* of K with real number coefficients, $C_i(K) = C_i(K, \mathbb{R})$, is a vector space over \mathbb{R} with basis $S_i(K)$. The i -th *cochain group* $C^i(K) = C^i(K, \mathbb{R})$ is the dual of the chain group which can be defined by $C^i(K) := \text{Hom}(C_i(K), \mathbb{R})$. Here $\text{Hom}(C_i, \mathbb{R})$ is the set of all homomorphisms of C_i into \mathbb{R} . For an $(i+1)$ -simplex $\sigma = [v_0, \dots, v_{i+1}]$, its *coboundary operator*, $\delta_i : C^i(K) \rightarrow C^{i-1}(K)$, is defined as

$$(\delta_i f)(\sigma) = \sum_{j=1}^{i+1} (-1)^j f([v_0, \dots, \hat{v}_j, \dots, v_{i+1}]),$$

where \hat{v}_j denotes that the vertex v_j has been omitted. The *boundary operators*, δ_i^* , are the adjoints of the coboundary operators,

$$\cdots C^{i-1}(K) \xrightleftharpoons[\delta_{i+1}^*]{\delta_{i+1}} C^i(K) \xrightleftharpoons[\delta_i^*]{\delta_i} C^{i+1}(K) \cdots$$

satisfying $(\delta_i a, b)_{C^{i+1}} = (a, \delta_i^* b)_{C^i}$ for every $a \in C^i(K)$ and $b \in C^{i+1}(K)$, where $(\cdot, \cdot)_{C^i}$ denote the scalar product on the cochain group.

In [15], the three *simplicial Laplacian operators* for higher-dimensional simplices, using the boundary and coboundary operators between chain groups, are defined as

$$\begin{aligned} \mathcal{L}_p^{\text{down}} &= \delta_{p-1} \delta_{p-1}^* && \text{down Laplacian} \\ \mathcal{L}_p^{\text{up}}(K) &= \delta_p^* \delta_p && \text{up Laplacian} \end{aligned}$$

$$\mathcal{L}_p(K) = \mathcal{L}_p^{\text{up}} + \mathcal{L}_p^{\text{down}} \quad \text{Laplacian}$$

These operators are self-adjoint, non-negative, compact and have different spectral properties [15].

To make the expression of Laplacian explicit, they identify each coboundary operator δ_p with an incidence matrix D_p in [15]. The *incidence matrix* $D_p \in \mathbb{R}_2^{n_{p+1}} \times \mathbb{R}_2^{n_p}$ encodes which p -simplices are incident to which $(p+1)$ -simplices where n_p is number of p -simplices. It is defined as

$$D_p(i, j) = \begin{cases} 1 & \text{if } \sigma_j^p \text{ is on the boundary of } \sigma_i^{p+1} \\ 0 & \text{otherwise} \end{cases}$$

Here, we assume the simplices are not oriented. One can incorporate the orientations by simply adding “ $D_p(i, j) = -1$ if σ_j^p is not coherent with the induced orientation of σ_i^{p+1} ” in the definition if needed.

Furthermore, we assume that the simplices are weighted, i.e., there is a weight function z defined on the set of all simplices of K whose range is \mathbb{R}^+ . Let W_p be an $n_p \times n_p$ diagonal matrix with $W_p(j, j) = z(\sigma_j^p)$ for all $j \in \{1, \dots, n_p\}$. Then, the i -dimensional up Laplacian can be expressed as the matrix

$$\mathcal{L}_i^{\text{up}} = W_i^{-1} D_i^T W_{i+1} D_i.$$

Similarly, the i -dimensional down Laplacian can be expressed as the matrix

$$\mathcal{L}_i^{\text{down}} = D_{i-1} W_{i-1}^{-1} D_{i-1}^T W_i.$$

$\mathcal{L}_i^{\text{down}}$ is only defined for $i \geq 1$ and is equal to 0 for $i = 0$. Then, to express the i -dimensional Laplacian \mathcal{L}_i , we can add these two matrices.

3 Hypergraph Laplacians

In this paper, we develop two new hypergraph Laplacians inspired from the simplicial Laplacian. The first hypergraph Laplacian allows defining diffusion between fixed dimension hyperedges *through* any hyperedges, and the second hypergraph Laplacian allows defining diffusion between *any* hyperedges through any hyperedges. In the simplicial Laplacian, a hyperedge of size $k+1$ is called a k -simplex. To be consistent with the simplicial Laplacian definition, we prefer to call a hyperedge of size $k+1$ as k -simplex in the rest of the paper.

3.1 A Hypergraph Laplacian Between Fixed Dimension Hyperedges

In this section, we present a new Laplacian between a fixed dimension simplices through any simplices. Let H be a hypergraph with the maximum simplex (hyperedge) dimension n . In the simplicial Laplacian in Sect. 2.2, the incidence matrix is only defined between p -simplices and $(p+1)$ -simplices for $0 \leq p < n$. In order to define the Laplacian between p -simplices through other simplices, not only $(p-1)$ - and $(p+1)$ -simplices, we define a new incidence matrix as follows.

Definition 1. The incidence matrix between p - and r -simplices $D_{p,r} \in \mathbb{R}_2^{n_r} \times \mathbb{R}_2^{n_p}$ for $p \leq r$ encodes which p -simplices are incident to which r -simplices where n_p is number of p -simplices. It is defined as

$$D_{p,r}(i,j) = \begin{cases} 1 & \text{if } \sigma_j^p \text{ is on the boundary of } \sigma_i^r \\ 0 & \text{otherwise} \end{cases}$$

The incidence matrix in the definition above allows us to define the Laplacian between k -simplices through any simplices as follows.

Definition 2. Laplacian between k -simplices through l -simplices in a hypergraph is defined as

$$\mathcal{L}_{k,l} = \begin{cases} W_l^{-1} D_{k,l}^T W_k D_{k,l} & \text{if } k \leq l \\ D_{l,k} W_k^{-1} D_{l,k}^T W_l & \text{if } k > l \end{cases}$$

In the definition above, we follow the idea of the up and down Laplacians defined in Sect. 2.2. Now, to define the hypergraph Laplacian between k -simplices through all simplices, we add up all the Laplacians as follows.

Definition 3. Let H be a hypergraph with the maximum simplex dimension n . Then, Laplacian between k -simplices through other simplices in H is defined as

$$\mathcal{L}_k = \mathcal{L}_{k,0} + \mathcal{L}_{k,1} + \cdots + \mathcal{L}_{k,n-1} + \mathcal{L}_{k,n}$$

for $k \in \{0, \dots, n\}$.

Here we provide an example to the hypergraph Laplacian in Definition 3 on a toy hypergraph.

Example 1. The hypergraph in Fig. 2 has four vertices (0-simplices), five edges (1-simplices) and two triangles (2-simplices).

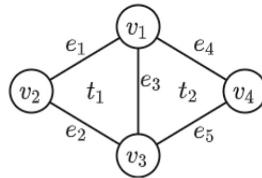


Fig. 2. A hypergraph with four vertices (0-simplices), five edges (1-simplices) and two triangles (2-simplices).

The corresponding incidence matrices as in Definition 1 are as follows.

$$D_{0,0} = I, D_{0,1} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}, D_{0,2} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}, D_{1,2} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Then, following Definitions 2 and 3, we get the Laplacian between 0, 1 and 2-simplices as follows.

$$\mathcal{L}_0 = \mathcal{L}_{0,0} + \mathcal{L}_{0,1} + \mathcal{L}_{0,2} = I + \begin{pmatrix} 3 & 1 & 1 & 1 \\ 1 & 2 & 1 & 0 \\ 1 & 1 & 3 & 1 \\ 1 & 0 & 1 & 2 \end{pmatrix} + \begin{pmatrix} 2 & 1 & 2 & 1 \\ 1 & 1 & 1 & 0 \\ 2 & 1 & 2 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 6 & 2 & 3 & 2 \\ 2 & 4 & 2 & 0 \\ 3 & 2 & 6 & 2 \\ 2 & 0 & 2 & 4 \end{pmatrix},$$

$$\mathcal{L}_1 = \mathcal{L}_{1,0} + \mathcal{L}_{1,1} + \mathcal{L}_{1,2} = \begin{pmatrix} 2 & 1 & 1 & 0 & 1 \\ 1 & 2 & 1 & 1 & 0 \\ 1 & 1 & 2 & 1 & 1 \\ 0 & 1 & 1 & 2 & 1 \\ 1 & 0 & 1 & 1 & 2 \end{pmatrix} + I + \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 2 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 4 & 2 & 2 & 0 & 1 \\ 2 & 4 & 2 & 1 & 0 \\ 2 & 2 & 5 & 2 & 2 \\ 0 & 1 & 2 & 4 & 2 \\ 1 & 0 & 2 & 2 & 4 \end{pmatrix},$$

$$\mathcal{L}_2 = \mathcal{L}_{2,0} + \mathcal{L}_{2,1} + \mathcal{L}_{2,2} = \begin{pmatrix} 3 & 2 \\ 2 & 3 \end{pmatrix} + \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix} + I = \begin{pmatrix} 7 & 3 \\ 3 & 7 \end{pmatrix}.$$

We can interpret the hypergraph Laplacians for a fixed dimension as follows. For a fixed dimension k , the diagonal entries show the number of neighboring simplices for each k -simplex. For example, in the toy graph, the vertex v_1 has six neighboring simplices as $v_1, e_1, e_3, e_5, t_1, t_2$ (we also count v_1 as a neighboring vertex in order to stress the importance of the direct neighborhood relation). That is why $\mathcal{L}_0(1, 1) = 6$. Furthermore, the off-diagonal entries show the number of shared neighboring simplices between k -simplices. For example, v_1 and v_2 share two neighboring simplices as e_1, t_1 . That is why $\mathcal{L}_0(1, 2) = \mathcal{L}_0(2, 1) = 2$.

3.2 A Generalized Hypergraph Laplacian

The hypergraph Laplacian in Definition 3 extends the simplicial Laplacian in a way to allow the diffusion through any simplices (hyperedges) between fixed dimensional simplices. However, this Laplacian is not able to capture the diffusion between different dimensional simplices. In order to define the generalized hypergraph Laplacian, we define a new incidence matrix that allows to encode the relation between p - and r -simplices through any simplices with $p < r$ as follows.

Definition 4. *Incidence matrix between p - and r -simplices through q simplices* $D_{p,r}^q \in \mathbb{R}_2^{n_r} \times \mathbb{R}_2^{n_p}$ *with $p < r$ encodes which p -simplices are incident to which r -simplices through q -simplices where n_p is number of p -simplices. For $q \notin \{p, r\}$, it is defined as*

$$D_{p,r}^q(i, j) = s$$

where s is the number of the q -simplices that are adjacent to both σ_j^p and σ_i^r . For $q \in \{p, r\}$, we take $D_{p,r}^p = D_{p,r}^r = D_{p,r}$ as in Definition 1.

Now, using the incidence matrix defined above, we define a new incidence matrix between p - and r -simplices through all simplices as follows.

Definition 5. Incidence matrix between p - and r -simplices through all simplices $\mathcal{D}_{p,r} \in \mathbb{R}_2^{n_r} \times \mathbb{R}_2^{n_p}$ with $p < r$ encodes which p -simplices are incident to which r -simplices through any simplex where n_p is number of p -simplices. It is defined as

$$\mathcal{D}_{p,r} = D_{p,r}^0 + D_{p,r}^1 + \cdots + D_{p,r}^n.$$

As the final step, we define the generalized hypergraph Laplacian between any simplices through any simplices as follows.

Definition 6. Let H be a hypergraph with the maximum simplex dimension n . Then we define the hypergraph Laplacian of H , \mathcal{L}_H , as the following block matrix

$$\mathcal{L}_H = \left(\begin{array}{c|c|c|c|c} \mathcal{L}_0 & \mathcal{D}_{0,1}^T & \mathcal{D}_{0,2}^T & \cdots & \mathcal{D}_{0,n}^T \\ \hline \mathcal{D}_{0,1} & \mathcal{L}_1 & \mathcal{D}_{1,2}^T & \cdots & \mathcal{D}_{1,n}^T \\ \hline \mathcal{D}_{0,2} & \mathcal{D}_{1,2} & \mathcal{L}_2 & \cdots & \mathcal{D}_{2,n}^T \\ \hline \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline \mathcal{D}_{0,n} & \mathcal{D}_{1,n} & \mathcal{D}_{2,n} & \cdots & \mathcal{L}_n \end{array} \right)$$

where $\mathcal{D}_{p,q}$ is the incidence matrix between p - and r simplices through all simplices of H and \mathcal{L}_k is the Laplacian between k -simplices through other simplices of H .

Here we continue the example in the previous section but this time show how to define the generalized hypergraph Laplacian.

Example 2. For the toy hypergraph in Fig. 2, the corresponding incidence matrices as in Definition 4 are

$$D_{0,1}^2 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 2 & 1 & 2 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}, D_{0,2}^1 = \begin{pmatrix} 2 & 2 & 2 & 0 \\ 2 & 0 & 2 & 2 \end{pmatrix}, D_{1,2}^0 = \begin{pmatrix} 2 & 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 2 & 2 \end{pmatrix}.$$

Then, the incidence matrices as in Definition 5 are

$$\mathcal{D}_{0,1} = \begin{pmatrix} 3 & 3 & 1 & 0 \\ 1 & 3 & 3 & 0 \\ 4 & 1 & 4 & 1 \\ 1 & 0 & 3 & 3 \\ 3 & 0 & 1 & 3 \end{pmatrix}, \mathcal{D}_{0,2} = \begin{pmatrix} 4 & 4 & 4 & 0 \\ 4 & 0 & 4 & 4 \end{pmatrix}, \mathcal{D}_{1,2} = \begin{pmatrix} 4 & 4 & 4 & 1 & 1 \\ 1 & 1 & 4 & 4 & 4 \end{pmatrix}.$$

Finally, if we combine these incidence matrices with the hypergraph Laplacians as in Definition 6, we get the generalized Laplacian for the hypergraph in Fig. 2 as follows

$$\mathcal{L}_H = \left(\begin{array}{ccc|ccc|c} 6 & 2 & 3 & 2 & 3 & 1 & 4 & 1 & 3 \\ 2 & 4 & 2 & 0 & 3 & 3 & 1 & 0 & 0 \\ 3 & 2 & 6 & 2 & 1 & 3 & 4 & 3 & 1 \\ 2 & 0 & 2 & 4 & 0 & 0 & 1 & 3 & 3 \\ \hline 3 & 3 & 1 & 0 & 4 & 2 & 2 & 0 & 1 \\ 1 & 3 & 3 & 0 & 2 & 4 & 2 & 1 & 0 \\ 4 & 1 & 4 & 1 & 2 & 2 & 5 & 2 & 2 \\ 1 & 0 & 3 & 3 & 0 & 1 & 2 & 4 & 2 \\ 3 & 0 & 1 & 3 & 1 & 0 & 2 & 2 & 4 \\ \hline 4 & 4 & 4 & 0 & 4 & 4 & 4 & 1 & 1 \\ 4 & 0 & 4 & 4 & 1 & 1 & 4 & 4 & 4 \\ \hline & & & & & & 7 & 3 \\ & & & & & & 4 & 0 & 7 \end{array} \right) .$$

As it happens in the previous hypergraph Laplacian, the diagonal entries show the number of neighboring simplices for each k -simplex and the off-diagonal entries show the number of the shared neighboring simplices with other simplices. The diffusion between simplices happens based on the number of the shared neighboring simplices with other simplices in the generalized Laplacian.

4 Experiments

In this section, we evaluate the proposed hypergraph Laplacians in Susceptible-Infected-Recovered (SIR) epidemic spreading model in hypergraphs. In the SIR model, each node is classified as a Susceptible node (S), Infected node (I), or a Recovered node (R) at any given moment. A selected node is initially infected, and the rest of the network is susceptible to be infected. In each propagation, the infected node can infect its neighboring hyperedges with probability μ . As this process is repeated, infected hyperedges can recover with probability β and are not susceptible to be infected again. The number of nodes that were infected in the hypergraph measures the diffusion level. In our experiments with the SIR models, we set infection rate based on μ_c , where $\mu_c = \frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle}$, as derived from [13], and $\langle k \rangle$ is equal to the average degree of the network (the average of number of edges the nodes in a dataset has). Furthermore, for the infection rate, we take the weights of interactions into consideration. In this paper, we set the infection rate of an interaction of weight $w > 0$ to $\mu_w = 1 - (1 - \mu)^w$ following [29]. For simplicity, the recovery rate is set as $\beta = 1$. The experiment is run 200 times for more reliable results. Since we only study the diffusion between nodes, we use \mathcal{L}_0 to model diffusion.

In this paper, we study the Enron email network as a case study. This network is an organizational network where the vertices are employees and hyperedges are comprised of the sender and all recipients of an email. It has 143 vertices and 1630 hyperedges [4]. This network has been studied widely, especially, to find influential employees in the organization [12, 28]. We study the same problem to show the effectiveness of one of the proposed hypergraph Laplacians in the diffusion framework.

For evaluation, we take each node as infected and the rest of the network as susceptible in each iteration. After running the experiment for a selected node

200 times, we take the average number of infected nodes in the network. A greater number means a greater spreading ability and a greater influence on diffusion. Table 1 includes top ten influential employees based on our experiments.

Table 1. Top ten influential employees with their designations at Enron based on our experiments.

1	Philip Allen	Manager
2	John Lavorato	CEO
3	Mike Grisby	Manager
4	David Delainey	CEO
5	Richard Shapiro	Vice President
6	James Steffes	Vice President
7	Scott Neal	Vice President
8	Louise Kitchen	President
9	Keith Holst	Director
10	Jeff Dasovich	Government Relation Executive

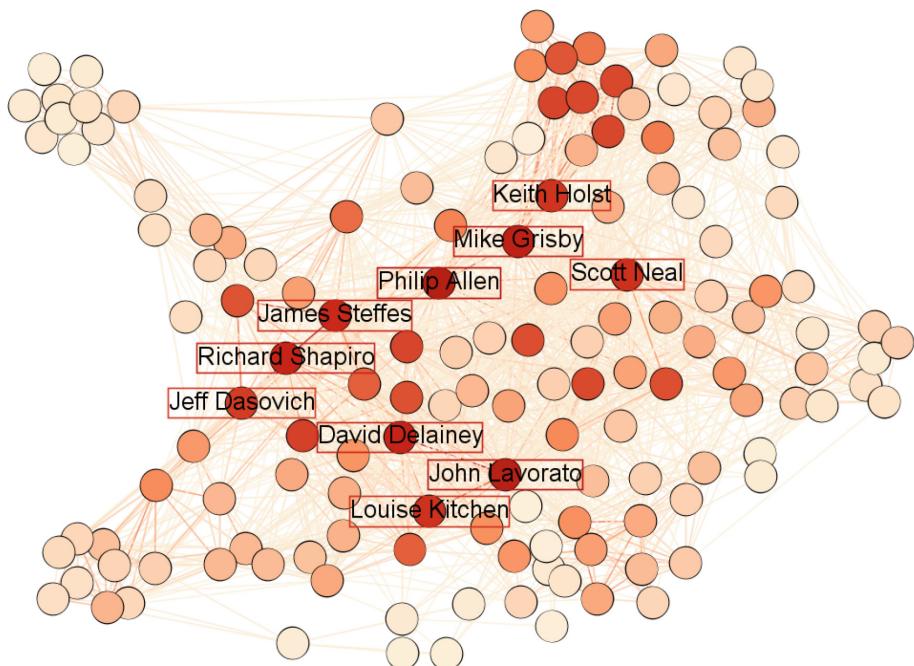


Fig. 3. The weighted Enron network. The darker edges have bigger weights and the darker nodes have a bigger number of infections. Labeled nodes are the top ten influential employees based on our experiments (see the detailed list in Table 1). More central nodes with larger edge weights have greater values, i.e., are more influential based on our experiments.

As we see in the table, the top ten influential list has the president, vice presidents, managers, a director, and a government relation executive. These are clearly influential designations in Enron. We also present the Enron network in Fig. 3, where nodes are colored based on the average number of infected people, and edges are colored based on the edge weight. Darker nodes and edges have bigger values. As we see in the figure, more central nodes with larger edge weights have greater values, i.e., are more influential based on our experiments. Hence, we can conclude that the proposed hypergraph Laplacian is effective in detecting influential nodes in hypergraphs.

5 Conclusion

In this paper, we develop two new hypergraph Laplacians based on the diffusion framework. We use the proposed Laplacians in the epidemic spreading model in hypergraphs, as an application. These Laplacians can be further employed in different network mining problems on hypergraphs, such as social contagion models on hypergraphs, influence study on hypergraphs, hypergraph classification, hypergraph neural networks, to list a few. As a future task, we plan to employ both Laplacians to find the influential higher-order structures in complex networks. As another future task, we plan to use both Laplacians in hypergraph representation learning.

References

1. Aggarwal, C.C., Wang, H.: Managing and Mining Graph Data, vol. 40. Springer, Heidelberg (2010). <https://doi.org/10.1007/978-1-4419-6045-0>
2. Aktas, M.E., Akbas, E., Fatmaoui, A.E.: Persistence homology of networks: methods and applications. *Appl. Netw. Sci.* **4**(1), 1–28 (2019). <https://doi.org/10.1007/s41109-019-0179-3>
3. Battiston, F., et al.: Networks beyond pairwise interactions: structure and dynamics. arXiv preprint [arXiv:2006.01764](https://arxiv.org/abs/2006.01764) (2020)
4. Benson, A.R., Abebe, R., Schaub, M.T., Jadbabaie, A., Kleinberg, J.: Simplicial closure and higher-order link prediction. *Proc. Natl. Acad. Sci.* **115**(48), E11221–E11230 (2018)
5. Carletti, T., Battiston, F., Cencetti, G., Fanelli, D.: Random walks on hypergraphs. *Phys. Rev. E* **101**(2), 022308 (2020)
6. Carletti, T., Fanelli, D., Lambiotte, R.: Random walks and community detection in hypergraphs. *J. Phys. Complex.* **2**(1), 015011 (2021)
7. Chitra, U., Raphael, B.: Random walks on hypergraphs with edge-dependent vertex weights. In: International Conference on Machine Learning, pp. 1172–1181. PMLR (2019)
8. Cook, D.J., Holder, L.B.: Mining Graph Data. Wiley (2006)
9. Cooper, C., Frieze, A., Radzik, T.: The cover times of random walks on random uniform hypergraphs. *Theoret. Comput. Sci.* **509**, 51–69 (2013)
10. Cooper, C., Dutle, A.: Spectra of uniform hypergraphs. *Linear Algebra Appl.* **436**(9), 3268–3292 (2012)

11. Dewar, M., et al.: Subhypergraphs in non-uniform random hypergraphs. arXiv preprint [arXiv:1703.07686](https://arxiv.org/abs/1703.07686) (2017)
12. Diesner, J., Frantz, T.L., Carley, K.M.: Communication networks from the Enron email corpus “it’s always about the people. Enron is no different”. *Comput. Math. Organ. Theor.* **11**(3), 201–228 (2005). <https://doi.org/10.1007/s10588-005-5377-0>
13. Guo, C., Yang, L., Chen, X., Chen, D., Gao, H., Ma, J.: Influential nodes identification in complex networks via information entropy. *Entropy* **22**(2), 242 (2020)
14. Hayashi, K., Aksoy, S.G., Park, C.H., Park, H.: Hypergraph random walks, Laplacians, and clustering. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 495–504 (2020)
15. Horak, D., Jost, J.: Spectra of combinatorial Laplace operators on simplicial complexes. *Adv. Math.* **244**, 303–336 (2013)
16. Hu, S., Qi, L.: The Laplacian of a uniform hypergraph. *J. Comb. Optim.* **29**(2), 331–366 (2015)
17. Huang, J., Zhang, R., Yu, J.X.: Scalable hypergraph learning and processing. In: 2015 IEEE International Conference on Data Mining, pp. 775–780. IEEE (2015)
18. Iacopini, I., Petri, G., Barrat, A., Latora, V.: Simplicial models of social contagion. *Nat. Commun.* **10**(1), 1–9 (2019)
19. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146 (2003)
20. Kirchhoff, G.: Ueber die auflösung der gleichungen, auf welche man bei der untersuchung der linearen vertheilung galvanischer ströme geführt wird. *Ann. Phys.* **148**(12), 497–508 (1847)
21. Kirkland, S.: Two-mode networks exhibiting data loss. *J. Complex Netw.* **6**(2), 297–316 (2018)
22. Li, M., Wang, X., Gao, K., Zhang, S.: A survey on information diffusion in online social networks: models and methods. *Information* **8**(4), 118 (2017)
23. Liu, Y., Yuan, J., Duan, B., Li, D.: Quantum walks on regular uniform hypergraphs. *Sci. Rep.* **8**(1), 1–8 (2018)
24. Lu, L., Peng, X.: High-order random walks and generalized Laplacians on hypergraphs. *Internet Math.* **9**(1), 3–32 (2013)
25. Merris, R.: Laplacian matrices of graphs: a survey. *Linear Algebra Appl.* **197**, 143–176 (1994)
26. Millán, A.P., Torres, J.J., Bianconi, G.: Explosive higher-order Kuramoto dynamics on simplicial complexes. *Phys. Rev. Lett.* **124**(21), 218301 (2020)
27. Mohar, B.: Some applications of Laplace eigenvalues of graphs. In: Hahn, G., Sabidussi, G. (eds.) *Graph Symmetry*. NATO ASI Series (Series C: Mathematical and Physical Sciences), vol. 497. Springer, Dordrecht (1997). https://doi.org/10.1007/978-94-015-8937-6_6
28. Shetty, J., Adibi, J.: Discovering important nodes through graph entropy the case of Enron email database. In: Proceedings of the 3rd International Workshop on Link Discovery, pp. 74–81 (2005)
29. Sun, Y., Liu, C., Zhang, C.X., Zhang, Z.K.: Epidemic spreading on weighted complex networks. *Phys. Lett. A* **378**(7–8), 635–640 (2014)
30. Wang, S., Gong, M., Liu, W., Wu, Y.: Preventing epidemic spreading in networks by community detection and memetic algorithm. *Appl. Soft Comput.* **89**, 106118 (2020)
31. Zhou, D., Huang, J., Schölkopf, B.: Learning with hypergraphs: Clustering, classification, and embedding. *Adv. Neural. Inf. Process. Syst.* **19**, 1601–1608 (2006)



Lumping Reductions for Multispread in Multi-Layer Networks

Tatjana Petrov^{1,2} and Stefano Tognazzi^{1,2(✉)}

¹ Universität Konstanz, Konstanz, Germany

{tatjana.petrov, stefano.tognazzi}@uni-konstanz.de

² Centre for the Advanced Study of Collective Behaviour, Konstanz, Germany

Abstract. Spreading phenomena arise from simple local interaction among a large number of actors through different networks of interactions. Computational modelling and analysis of such phenomena is challenging due to the combinatorial explosion of possible network configurations. Traditional (single layer) networks are commonly used to encode the heterogeneous relationships among agents but are limited to a single type of interaction. Multiplex Multi-Layer networks (MLNs) have been introduced to allow the modeler to compactly and naturally describe multiple types of interactions and multiple simultaneous spreading phenomena. The downside is an increase in the complexity of the already challenging task of the analysis and simulation of such spreading processes. In this paper we explore the use of lumping techniques that preserve dynamics, previously applied to Continuous Time Markov Chains (CTMC) and single layer networks to multiple spreading processes on MLNs.

Keywords: Multiplex multi-layer networks · Spreading processes · Model reduction techniques · Lumping · Stochastic processes

1 Introduction

Spreading phenomena such as epidemics emerge from simple, local interactions among a large number of actors, influencing each-other through different networks of interaction. The ability to faithfully model and predict the macroscopic consequences of spreading phenomena over networks, is of key importance in a wide range of application scenarios, ranging from mitigating epidemics [14], to understanding animal collectives [7] and online social networks [20] to cite a few.

Computational modelling and analysis of spreading processes quickly becomes challenging, due to the combinatorial explosion of possible network configurations, typically evolving stochastically over time, resulting into a large-scale continuous-time Markov chain (CTMC) [2, 9, 19]. For instance, a network of n actors where each actor can either be infected (I) or susceptible (S) to infection, gives rise to 2^n possible network configurations. On the other hand, *Reaction network* formalism facilitates the description of actors interacting based on their feature described by a suitable variable name. For example, a Susceptible-Infected-Susceptible (SIS) model, widely used to study the spread of opinions, rumours and memes in social networks, is specified through two

local interactions between actors in state S or I : (i) $S + I \rightarrow 2I$ for infection spread and (ii) $I \rightarrow S$ for recovery. Such model typically does not specify whether two actors are related or not and hence assumes population homogeneity. Homogeneity assumption allows to reduce the number of states through a *population abstraction*, requiring to only enumerate the total number of actors in each of the states (S and I), hence reducing the number of states in the above example from 2^n to n . Network homogeneity is however a strong assumption in most real-world scenarios. For instance, in case of epidemic spread, different individuals will have a different range and intensity of interactions with family, friends and coworkers that form a network of physical contact in which an infection can spread. State representation in form of a single-layered network configuration (where each actor has a specific state, e.g. S or I) allows to encode such relational heterogeneity. Reaction network formalism can facilitate such encoding through identifiers denoting position of actors in a network, e.g. $S_i + I_j \rightarrow I_i + I_j$, and conditioning the respective rate to the existence of an edge between actors [2, 8, 9]. Moreover, in reality, the same group of individuals can partake in different spreading processes at different interaction networks, which in turn affect each-other [1]. In these cases, independent analysis of spreading processes over single-layer networks is limited. To exemplify, while the infection spreads through physical contact, the rate of contact is influenced by other factors, such as awareness: an agent aware of the disease will have less contact; In turn, an agent with the disease will spread awareness more actively. Awareness will spread as well, through a communication network that may have significantly different dynamics than the infection spread. A faithful representation of interrelated spreading processes will further blow-up the space of states and parameters in a model [8]. Unlike single-layer networks, representing system state in terms of a multi-layer network (MLNs) allows to simultaneously incorporate multiple layers of relationship between network actors, as well as inter-layer correlations, in a natural and compact way [5, 8, 12, 15]. However, more detailed description further challenges the respective computational simulation and analysis. Formal reductions based on lumping states that are behaviourally equivalent are desirable [17], yet novel techniques are needed for the context of MLNs. In this paper, we propose a number of formal model reduction techniques for MLNs. The techniques are inspired by reductions detecting symmetries that aim to provably preserve the properties of the original system through lumping states. Different state representations and respective semantics of executions are subject to different lumping techniques, including the state lumping ideas previously used in context of reductions of CTMCs and differential-drift dynamical systems (preserving dynamical features), as well as reductions of static, undirected networks (preserving structural properties such as network centrality). We then empirically demonstrate and compare the performance of different reductions over a variety of artificially generated and real-world MLNs. Finally, we show how to efficiently compute the proposed reductions, and we show how to speed up the respective computational simulation of complex spreading processes on MLNs.

2 Background

Notation. Throughout this work, when clear from context, we will use x_i both to denote the i -th element of vector \boldsymbol{x} or the value of the map $x(i)$. For a partition \mathcal{H} over a

variable set $V_p \subseteq \{\mathbf{x}_1, \mathbf{x}_2, \dots\}$, induced by an equivalence relation $\sim_{\mathcal{H}} \subseteq V_p \times V_p$, we will denote elements of a partition class $H \in \mathcal{H}$ by $\mathbf{x}_{H,1}, \mathbf{x}_{H,2}, \dots, \mathbf{x}_{H,|H|}$. We denote by $\|\cdot\|_1$ the 1-norm. We will denote with $V_N = \{1, \dots, N\}$, $V_L = \{1, \dots, L\}$ the set of actors and layers, respectively. We will assume $N > 0$ and $L > 1$ (MLNs with $L = 1$ are a special case that corresponds with traditional single-layer networks). Vectors will be assumed to be written in column notation.

2.1 Reaction Networks

A reaction network is formally a pair (S, R) where S is a set of species and R is a set of reactions. Each reaction is in the form $\rho \xrightarrow{\alpha} \pi$, where $\alpha > 0$ is a kinetic parameter and ρ and π are multisets of species called reactants and products, respectively. The multiplicity of species S in ρ is denoted with $\rho(S)$, which represents the stoichiometry coefficient. The set of all reagents and products across all the reactions in the network are denoted by $\rho(R)$ and $\pi(R)$. Throughout this work, we will consider the *stochastic* semantics. It is worth mentioning when we refer to the deterministic semantics of reaction networks we use Ordinary Differential Equations with mass-action kinetics.

Stochastic Semantics. The stochastic semantics of a reaction network is given by a Continuous Time Markov Chain (CTMC) where each state σ is a multiset of species. From a state σ such that $\rho \subset \sigma$, a reaction $\rho \xrightarrow{\alpha} \pi$ induces a transition with mass-action propensity $\alpha \prod_{S \in \rho} \binom{\sigma(S)}{\rho(S)}$ to state $\sigma + \pi - \rho$, where the plus and minus operators indicate multiset union and difference, respectively, while $S \in \rho$ denotes that S belongs to the support of $\rho(S)$, i.e. $\rho(S) > 0$. Given an initial state $\hat{\sigma}$, the state space can be derived by exhaustively applying the reactions to compute all possible states reachable from $\hat{\sigma}$. We denote $out(\sigma)$ the multiset of outgoing transitions from state σ ,

$$out(\sigma) = \{\sigma \xrightarrow{\lambda} \sigma + \pi - \rho \mid (\rho \xrightarrow{\alpha} \pi) \in \mathcal{R}, \lambda = \alpha \prod_{S \in \rho} \binom{\sigma(S)}{\rho(S)}\}$$

For any two distinct states σ and ϕ , we denote by $q(\sigma, \phi)$ the sum of the propensities from σ to ϕ across all the reactions, that is

$$q(\sigma, \phi) = \sum_{(\sigma \xrightarrow{\lambda} \phi) \in out(\sigma)} \lambda$$

Moreover, we set $q(\sigma, \sigma)$ to be the negative sum of all possible transitions from state σ , i.e., $q(\sigma, \sigma) = -\sum_{\phi \neq \sigma} q(\sigma, \phi)$. These values ensure a well-formed CTMC generator matrix, which characterises the dynamical evolution of the CTMC. Each component of its solution, is the probability of being in a given multiset of species at time t starting from some initial probability distribution.

2.2 Multiplex Multi-Layer Networks

In this paper, we use a generalisation of networks called *multiplex networks* or *edge-colored-graphs*, which are useful for simultaneously representing different kinds of relationships over the same set of actors [5]. This paper will focus on *undirected* multiplex networks.

Definition 1. A multiplex network with N actors and L layers is an ordered collection of L undirected graphs over the same set of actors:

$$\mathcal{G} = \{G^{(l)} = (V_N, E^{(l)})\}_{l \in V_L},$$

where $E^{(l)} : V_N \times V_N \rightarrow \mathbb{R}_{\geq 0}$ are the edges on layer $l \in V_L$. For every layer l , we denote the non-negative adjacency matrix of the graph $G^{(l)}$ by $\mathbf{A}^{(l)} = (A_{ij}^{(l)}) \in \mathbb{R}_{\geq 0}^{N \times N}$. Then, the multiplex network can be represented by a 3rd-order adjacency tensor:

$$\mathcal{A} = (\mathcal{A}_{ijl}) \in \mathbb{R}_{\geq 0}^{N \times N \times L}, \text{ such that } \mathcal{A}_{ijl} := A_{ij}^{(l)} = E^{(l)}(i, j),$$

that is, \mathcal{A}_{ijl} represents the presence of an edge between actors i and j on layer l .

2.3 Lumping Species in a Reaction Network

We next review three formal reductions techniques for lumping species, based on the reaction network description [3]. Each of the techniques was proposed with a goal to guarantee a certain semantic relationships. These reduction ideas will be employed for reducing spreading processes over MLNs. Let (S, R) be the reaction network. Then,

- (Forward Equivalence) $\sim_{FE} \subseteq S \times S$ is a forward equivalence, if sum of the drift functions in the respective differential semantics for any two equivalent states is equivalent (up to \sim_{FE}). The condition guarantees that the sum of solutions for species lumped by \sim_{FE} will be equal to the solution of respective macro-species in the reduced ODE system. Given a reaction network, finding relation \sim_{FE} can be done in polynomial time¹ [3].
- (Backward Equivalence) On the other hand, $\sim_{BE} \subseteq S \times S$ if the drift functions in the respective differential semantics for any two equivalent states are equivalent (up to \sim_{BE}). The lumping condition guarantees that, in case two lumped species start from the same initial conditions, their solutions in the deterministic semantics will coincide across time. The complexity of finding \sim_{BE} is the same as for \sim_{FE} .
- (Stochastic Equivalence) $\sim_{SE} \subseteq S \times S$ is a stochastic equivalence, where two species are lumped, if the cumulative rates towards any partition of multi-sets inherited by \sim_{SE} from any multi-set containing s and resp. s' are equal. This cumulative rate will represent the rate between the respective partitions of multi-sets in the reduced model. The condition guarantees that the partition over the CTMC states inherited from the partitioning over the species set (\sim_{SE}) will be ordinary lumpable [2]. Finding the partition \sim_{SE} is polynomial in the size of reaction network $O(|R||S|\log|S|)$. This significantly improves the complexity of searching for the lumpable partition directly over the expanded CTMC.

¹ The algorithm is a variation of the Paige-Tarjan algorithm.

2.4 Lumping Actors in Multiplex Networks

Lumping techniques can be used as an efficient algorithmic procedure to compute structural properties of actors within a network. In recent works [15], we use formal reductions to compute one notion of eigenvector centrality for multiplex MLNs, proposed in [18]. The centrality is defined through a 2-map, *f-eigenvector centrality*, in which the first component of the map represents the centrality associated to the *actors*, while the second component is centrality associated to the *layers*.

Definition 2 ([18]). Let $\mathcal{A} \in \mathbb{R}_{\geq 0}^{N \times N \times L}$ be the adjacency tensor of an MLN with weighted, undirected layers, and let $\alpha, \beta > 0$ be such that $\frac{2}{\beta} < (\alpha - 1)$. Then, define $f = (f_1, f_2) : \mathbb{R}_{\geq 0}^N \times \mathbb{R}_{\geq 0}^L \rightarrow \mathbb{R}_{\geq 0}^N \times \mathbb{R}_{\geq 0}^L$ as follows:

$$f_1(\mathbf{x}, \mathbf{t})_i = \left(\sum_{j=1}^N \sum_{l=1}^L A_{ijl} x_j t_l \right)^{\frac{1}{\alpha}} \quad \text{for } i \in V_N, \quad f_2(\mathbf{x}, \mathbf{t})_l = \left(\sum_{i=1}^N \sum_{j=1}^N A_{ijl} x_i x_j \right)^{\frac{1}{\beta}} \quad \text{for } l \in V_L.$$

In words, the centrality x_i of an actor i is a sum of the centralities of each of its neighbouring actors, weighted by the product of the edge-weight and the centrality of the layer at which that connection lies. The parameters α and β are introduced in order to guarantee convergence and respectively well-definedness in case of undirected MLNs.² In [18] the centrality vector of the actors and layers is denoted by $(\mathbf{x}^*, \mathbf{t}^*) \in \mathbb{R}_{\geq 0}^N \times \mathbb{R}_{\geq 0}^L$ which is a limit of an iterative scheme based on the 2-map f .

- (Actor Equivalence) $\sim_{AE} \subseteq V_N \times V_N$ is an actor equivalence, where two actors are lumped, if they have the same *f*-eigenvector centrality value (i.e., $x_i \sim_{AE} x_j$, if $x_i^* = x_j^*$). Finding the partition \sim_{AE} is polynomial in the size of the MLN $O(|E| \log(|V_N| + |V_L|))$ where $|E|$ denotes the total number of edges in all the layers [15].

Related Works. In [11] the authors propose a unified taxonomy of MLN simplification techniques. It is worth mentioning that *FE*, *BE* and *SE* can not be considered as explicit MLN simplification techniques, because they are acting directly at the level of the *Reaction Network*. However, *AE* is a MLN simplification, because it is acting at the level of actors (it is an *aggregation* technique based on *positional equivalence*).

3 Results

3.1 Interacting Spreading Processes on Multiplex Multi-Layer Networks

In this work we focus on interacting spreading processes, sometimes referred as *multispread* processes.

Definition 3 Let \mathcal{G} be a multiplex. A multispread $\mathcal{MS} = (\mathcal{G}, IS, P, IC, SR, LIR)$ is a tuple composed by:

² Further discussion on the choice of α and β is beyond the scope of this manuscript and we refer the interested reader to [18].

- A multiplex \mathcal{G} with N actors and L layers;
- A set of internal states $IS = \{IS_1, \dots, IS_M\}$;
- A set of rates $P = \{r_1, \dots, r_R\}$ s.t. for all $i \in \{1, \dots, R\}$, $r_i \in \mathbb{R}^+$;
- A set of initial conditions $IC : \{1, \dots, N\} \rightarrow IS$;
- A set of single actor rules SR . SR rules are in the form $ISx \xrightarrow{r} ISy$, where $ISx, ISy \in IS$ and $r \in P$.
- A set of local interaction rules LIR . LIR rules are in the form $ISx \xrightarrow{l} ISy \xrightarrow{r} ISy$ where $ISx, ISy \in IS$, $l \in \{1, \dots, L\}$ and $r \in P$.

Where $ISx \xrightarrow{l} ISy$ represents the existence of an edge in layer l between agents that are in internal state ISx and ISy respectively.

Definition 4 (Compiling a multispread into a RN)

Let $\mathcal{MS} = (\mathcal{G}, IS, P, IC, SR, LIR)$ be a multispread process, the resulting Reaction Network $RN_{\mathcal{MS}} = (S, R)$ is constructed as follows:

- The set of species S is the union, for all $i \in \{1, \dots, N\}$ of the set of all the internal states of each actor $Act_i = \{IS_{1i}, \dots, IS_{Mi}\}$.
- The set of reactions $R = R_{SR} \cup R_{LIR}$ is the union of the reactions that we compile from the single actor rules R_{SR} and the local interaction rules R_{LIR} .
- For each rule $sr \in SR$ where sr is in the form $ISx \xrightarrow{r} ISy$ we build the set of reactions $R_{sr} = \{ISx_i \xrightarrow{r} ISy_i \mid i \in \{1, \dots, N\}\}$
- For each rule $lir \in LIR$ where lir is in the form $ISx \xrightarrow{l} ISy \xrightarrow{r} ISy \xrightarrow{l} ISy$ we build the set of reactions $R_{lir} = \{ISx_i + ISy_j \xrightarrow{r} ISy_i + ISy_j \mid (i, j) \in E^{(l)}\}$.
- We set the following initial conditions:

$$ISx_i = \begin{cases} 1 & \text{if } IC(i) = ISx, \\ 0 & \text{if } IC(i) \neq ISx. \end{cases}$$

3.2 The Multispread Model

In this work, we evaluate the performance of the lumping techniques reviewed in Sect. 2 to multispreads that arise from MLNs. We will use two different sets of benchmarks: real-world networks from the Koblenz Network Collection [13], and a set of synthetic networks. The synthetic networks will have a physical layer (referred as layer 1) built using a power-law degree distribution network generated with a configuration model with exponent 2.5. In all case studies, the virtual layer (referred as layer 2) is a copy of the physical layer network with an added percentage of random edges (non-overlapping with previous edges). We showcase our findings with the aim of obtaining the maximal aggregation, if not specified differently. The aim of the *maximal aggregation* approach is to obtain the smallest possible reduced system. In this work we use the following multispread inspired by the interacting spreading processes presented in [8].

$$IS = \{US, AS, AI\}$$

$$P = \{\delta = 0.6, \mu = 0.4, \beta^A = 0.01, \beta^U = 0.4, \lambda = 0.15\}$$

$$SR = \{AS \xrightarrow{\delta} US, AI \xrightarrow{\mu} AS\}$$

$$\begin{aligned} LIR = & \{AS \xrightarrow{1} AI \xrightarrow{\beta^A} AI \xrightarrow{1} AI, US \xrightarrow{1} AI \xrightarrow{\beta^U} AI \xrightarrow{1} AI, \\ & US \xrightarrow{2} AI \xrightarrow{\lambda} AS \xrightarrow{2} AI, US \xrightarrow{2} AS \xrightarrow{\lambda} AS \xrightarrow{2} AS\} \end{aligned}$$

We consider non-degenerate reductions using initial partitions with 3 blocks: $\{AI_1, \dots, AI_N\}$, $\{AS_1, \dots, AS_N\}$ and $\{US_1, \dots, US_N\}$.

3.3 Experimental Setup

All the experiments presented in this paper rely on three components. First, a Python script using the *Networkx* package [10] is used to generate the synthetic Power-Law MLNs. The second step is comprised of MATLAB scripts which parse instances into a series of models for ERODE [4]. ERODE, a state-of-the-art model reduction tool, is used to compute Backward Equivalence, Forward Equivalence, Stochastic Equivalence and Actor Equivalence. ERODE provides the following outputs: the partitions that were computed and the reduced models. We use the state-of-the-art tool *StochKit* [16] to run the stochastic. All experiments have been conducted on a MacBook Pro with a 2.6 GHz Intel Core i7 with 16 GB of RAM.³ Throughout this Section we will use *PL-x* to refer to the synthesised MLNs with x being the number of actors. In all the experiments that involve runs of the stochastic simulations we used time horizon $T = 10$.

3.4 Size of the Reduction

In this set of experiments, we compare the size of the obtained reductions using BE, FE and SE, applied to the MLN by translating it to a reaction network formalism, using Definition 4. The results are presented in Table 1. For each instance, we report the percentage of added edges in the virtual layer, the number of species $|S|$ of the Reaction Network of the original model (i.e., $|S| = 3 \cdot N$, where N is the number of actors), the number of species of the reduced model via BE and the reduction ratio (number of species in the reduced model, divided by the number of species in the original model). Analogously, in the last three columns we present the size of the reductions obtained with FE, SE, as well as their reduction ratio.

First, we notice that FE and SE compute the same partitions: this is because FE and SE both characterise ordinary lumpability and in the case of the UAU-SIS spreading process that we consider in this work they coincide.

Secondly, we can notice how BE and FE are notions that are not comparable with each other, as discussed in [2, 6].

In the presented instances, adding more edges results in more refinement. This is expected to happen because most of the techniques presented exploit symmetries

³ The code and examples are available <https://github.com/stefanotognazzi/LumpingForMLNs>.

and the act of adding more edges usually leads to a smaller amount of symmetries in the model. Ideally, the smaller the reduction ratio the better but, when dealing with instances that arise from real-world scenarios, it is known that it is rare to find significant reductions due to their highly non-symmetrical nature. However, we will later show in Sect. 3.6 that, even with the reduction ratios presented in Table 1, we obtain significant speed-ups in computing stochastic simulations.

Table 1. Size of reductions

Maximal Aggregation			Summary of reductions				Maximal Aggregation			Summary of reductions				
Instance	Added	S	S (BE)	BE Ratio	S (FE/SE)	FE/SE ratio		Instance	Added	S	S (BE)	BE Ratio	S (FE/SE)	FE/SE ratio
PL-100	5	300	255	85.0%	264	88.0%		PL-10000	5	30000	22497	75.0%	25332	84.4%
PL-100	10	300	270	90.0%	282	94.0%		PL-10000	10	30000	24138	80.5%	26268	87.6%
PL-500	5	1500	1203	80.2%	1302	86.8%		EgoFB	5	8664	420	4.8%	939	10.8%
PL-500	10	1500	1296	86.4%	1389	92.6%		EgoFB	10	8664	759	8.8%	1659	19.1%
PL-1000	5	3000	2475	82.5%	2706	90.2%		As2000	5	19422	13017	67.0%	13455	69.3%
PL-1000	10	3000	2553	85.1%	2763	92.1%		As2000	10	19422	14580	75.1%	14853	76.5%
PL-5000	5	15000	11883	79.2%	13137	87.6%		PGP	5	32040	26445	82.5%	27708	86.5%
PL-5000	10	15000	12519	83.5%	13509	90.1%		PGP	10	32040	28221	88.1%	28992	90.5%

3.5 Cost of the Reduction

In this set of experiments, we show the computational cost (in terms of time) of obtaining the reductions. Results are summarised in Table 2. We show for each instance the number of species in the original model ($|S|$) and for each of the proposed techniques the time (in seconds) required by ERODE to obtain the partitions presented in Table 1. BE and FE are computationally efficient. SE is polynomial but, because of the added constraints, in practice it is more computationally costly.

Table 2. Time of reductions

Instance	Added	S	BE(s)	FE(s)	SE(s)		Instance	Added	S	BE(s)	FE(s)	SE(s)
PL-100	5	300	0.004	0.005	0.029		PL-10000	5	30000	0.638	0.687	304.188
PL-100	10	300	0.005	0.005	0.035		PL-10000	10	30000	0.663	0.797	289.318
PL-500	5	1500	0.027	0.033	0.400		EgoFB	5	8664	0.098	0.120	1.145
PL-500	10	1500	0.028	0.034	0.470		EgoFB	10	8664	0.097	0.175	2.092
PL-1000	5	3000	0.066	0.070	1.624		As2000	5	19422	0.399	0.632	96.712
PL-1000	10	3000	0.096	0.128	2.447		As2000	10	19422	0.471	0.733	175.666
PL-5000	5	15000	0.289	0.349	59.171		PGP	5	32040	0.884	1.073	409.404
PL-5000	10	15000	0.290	0.399	87.038		PGP	10	32040	0.942	1.043	498.678

3.6 Speeding up Stochastic Simulations

In this set of experiments, we show the benefits of using the reduced models in terms of the speed-up of the stochastic simulations. The results are summarised in Table 3. We conduct stochastic simulations using SSA. All the results are presented in seconds and the reported time of one run is obtained as the time of a run averaged over a repetition of 5 runs. For each instance, we show the time (in seconds) of computing one run of SSA on the full model. In the middle columns, we report the time of computing one run of SSA algorithm for BE, FE and SE reduced models with maximal aggregation. This can be explained by the fact that the run-time of simulation is superlinear wrt. actor count.

Table 3. Time of simulation

Time for one SSA run (s)						Time for one SSA run (s)						
Instance	Added	Full	BE	FE	SE		Instance	Added	Full	BE	FE	SE
PL-100	5	0.066	0.058	0.064	0.064		PL-10000	5	514.460	193.087	230.441	239.246
PL-100	10	0.064	0.059	0.063	0.062		PL-10000	10	535.360	232.286	264.743	267.900
PL-500	5	0.333	0.269	0.289	0.284		EgoFB	5	48.002	0.102	0.259	0.264
PL-500	10	0.335	0.290	0.311	0.313		EgoFB	10	48.194	0.192	0.742	0.793
PL-1000	5	0.916	0.745	0.826	0.830		As2000	5	215.274	55.684	64.782	66.634
PL-1000	10	0.934	0.787	0.842	0.867		As2000	10	214.124	80.212	87.279	87.460
PL-5000	5	12.873	9.395	10.304	10.317		PGP	5	68.481	51.766	56.884	59.885
PL-5000	10	13.057	10.042	10.938	12.311		PGP	10	70.543	58.104	63.297	65.535

3.7 Approximation of the Reduction

In this set of experiments we aim at providing experimental evidence that, despite the fact that BE in general is an approximation of the stochastic semantics, we can use that reduction in this scenario as a good approximation of the original solution. We use synthetised MLNs with 15% of added edges in the virtual layer with a number of actors ranging from 20 to 200. In order to obtain precision in the solution we computed 1 Million runs of SSA on each of the presented instances. Accuracy results are presented with respect to the simulated number of actors in each state and compared to the solution of the original instance. In this set of experiments we fix an initial partition based on the initial conditions proposed in [8]. In Table 4 we present the average across all runs of the number of actors in each state at the end of the simulation. In Table 5 we show the maximum error, in terms of percentage of distance to the solution of the original instance, that we could observe at all time points. From [2] we know that the reduction obtained via FE/SE is exact in the sense that it can be used to replicate exactly the stochastic semantics of the original model. Therefore, we report the results obtained using BE.

3.8 Reduction at the Level of Actors

In this set of experiments we show how to obtain the same partition obtainable with BE lumping at the level of the reaction network by using Actor Equivalence (AE) from

Sect. 2.4 that acts at the level of the MLN’s actors. We show experimental evidence that the two reductions coincide when interpreted over the actors. The results are summarised in Table 6. We report the results for the Power-Law MLNs with 5% of added edges. To facilitate the interpretation of the results we provide in the table, alongside the number of original species and reduced species, the size in terms of actors of the MLNs of the obtained partitions.

The advantage of this approach is that Actor Equivalence reduces a model that has a number of species that is $|V| + |L|$, while, using BE on the reaction network we need to reduce a system such that $|S| = 3 \cdot |V|$. From a theoretical perspective this fact does not yield any improvement in terms of complexity but we show in Table 6 that in practice we obtain a speed up and the ability to scale to larger size networks.

Table 4. Approximations of reduction

Instance	Accuracy			Instance	Accuracy		
	AI	US	AS		PL-70	AI	US
PL-20	1.0516	17.0271	1.9213	Full	14.6530	33.7650	21.5820
BE	1.0532	17.0223	1.9245	BE	14.6513	33.7610	21.5877
PL-30	AI	US	AS	PL-80	AI	US	AS
Full	3.2101	21.5280	5.2620	Full	12.7763	47.6297	19.5940
BE	3.2082	21.5209	5.2709	BE	12.7791	47.6390	19.5819
PL-40	AI	US	AS	PL-90	AI	US	AS
Full	2.6129	32.8668	4.5204	Full	11.6084	59.9792	18.4124
BE	2.6187	32.8548	4.5265	BE	11.5948	59.9885	18.4167
PL-50	AI	US	AS	PL-100	AI	US	AS
Full	7.5197	30.8025	11.6778	Full	19.5981	52.0239	28.3779
BE	7.5192	30.8018	11.6790	BE	19.6470	51.9174	28.4356
PL-60	AI	US	AS	PL-200	AI	US	AS
Full	7.4002	41.0514	11.5483	Full	27.2781	130.9980	41.7239
BE	7.3935	41.0569	11.5496	BE	27.2851	130.9977	41.7172

Table 5. Max error

Instance	BE
PL-20	0.36%
PL-30	0.17%
PL-40	0.22%
PL-50	0.04%
PL-60	0.16%
PL-70	0.05%
PL-80	0.06%
PL-90	0.15%
PL-100	0.25%
PL-200	0.03%

Table 6. Actor Equivalence, 5% added edges

Maximal aggregation		Actor Lumping				Species Lumping			
Instance		S	S (AE)	Actors (AE)	Time (s)	S full	S (BE)	Actors (BE)	Time (s)
PL-1000		1002	827	825	0.028	3000	2475	825	0.096
PL-5000		5002	3963	3961	0.124	15000	11883	3961	0.300
PL-10000		10002	7501	7499	0.369	30000	22497	7499	0.656
PL-50000		50002	38027	38025	1.818	150000	114075	38025	3.896
PL-100000		100002	77489	77487	3.921	300000	232461	77487	7.388
PL-500000		500002	382068	382066	24.049	1500000	1146198	382066	59.850
PL-1000000		1000002	776457	776455	56.244	3000000	—	O.O.M.	—

4 Conclusions and Future Works

Stochastic semantics are a key tool to understand and study spreading processes in networked systems. Analysing interacting spreading processes on complex multiplex Multi-Layer Networks is computationally costly, if feasible at all. In this work, we extended a variety of lumping-based automated model reduction techniques to interacting spreading processes on Multiplex Multi-Layer Networks that allows the modeler to run the stochastic simulations at a cheaper computational cost. Our findings show experimental evidence that in the context of multispread processes over MLNs, efficient reduction techniques originally designed to exactly preserve differential semantics, faithfully abstract the stochastic semantics. In future work, we plan to investigate these results from a theoretical standpoint, that will set foundations for exploiting this scalable approach in practice.

Acknowledgements. This work was supported by the Ministry of Science, Research and the Arts of the state of Baden-Württemberg, and the DFG Centre of Excellence 2117 ‘Centre for the Advanced Study of Collective Behaviour’ (ID: 422037984). The authors would like to thank Giacomo Rapisardi for the inspiring discussions on the topic and Andrea Vandin for the support and the insights on the use of the tool ERODE.

References

1. Bródka, P., Musial, K., Jankowski, J.: Interacting spreading processes in multilayer networks: a systematic review. *IEEE Access* **8**, 10316–10341 (2020)
2. Cardelli, L., Perez-Verona, I.C., Tribastone, M., Tschaikowski, M., Vandin, A., Waizmann, T.: Exact maximal reduction of stochastic reaction networks by species lumping. *Bioinformatics* (2021)
3. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Symbolic computation of differential equivalences. In: 43st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL) (2016)
4. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: ERODE: a tool for the evaluation and reduction of ordinary differential equations. In: TACAS, pp. 310–328 (2017)
5. De Domenico, M., et al.: Mathematical formulation of multi-layer networks. *Phys. Rev. X* **3**, 07 (2013)
6. Feret, J., Henzinger, T., Koeppl, H., Petrov, T.: Lumpability abstractions of rule-based systems. *Theor. Comput. Sci.* **431**, 137–164 (2012)
7. Finn, K., Silk, M., Porter, M., Pinter-Wollman, N.: The use of multilayer network analysis in animal behaviour. *Anim. Behav.* **149**, 7–22 (2019)
8. Granell, C., Gomez, S., Arenas, A.: Dynamical interplay between awareness and epidemic spreading in multiplex networks. *Phys. Rev. Lett.* **111**, 128701 (2013)
9. Grossmann, G., Bortolussi, L.: Reducing spreading processes on networks to Markov population models, pp. 292–309 (2019)
10. Hagberg, A.A., Schult, D.A., Swart, P.J.: Exploring network structure, dynamics, and function using networkx. In: Varoquaux, G., Vaught, T., Millman, J. (eds.) Proceedings of the 7th Python in Science Conference, Pasadena, CA USA, pp. 11–15 (2008)
11. Interdonato, R., Magnani, M., Perna, D., Tagarelli, A., Vega, D.: Multilayer network simplification: approaches, models and methods. *Comput. Sci. Rev.* **36**, 100246 (2020)

12. Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J.P., Moreno, Y., Porter, M.A.: Multilayer networks. *J. Complex Netw.* **2**(3), 203–271 (2014)
13. Kunegis, J.: KONECT - the koblenz network collection. In: Proceedings of International Conference on World Wide Web Companion, pp. 1343–1350 (2013)
14. Pastor-Satorras, R., Castellano, C., Van Mieghem, P., Vespignani, A.: Epidemic processes in complex networks. *Rev. Mod. Phys.* **87**, 925–979 (2015)
15. Petrov, T., Tognazzi, S.: Centrality-preserving exact reductions of multi-layer networks. In: Margaria, T., Steffen, B. (eds.) ISoLA 2020. LNCS, vol. 12477, pp. 397–415. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-61470-6_24
16. Sanft, K.R., et al.: StochKit2: software for discrete stochastic simulation of biochemical systems with events. *Bioinformatics* **27**(17), 2457–2458 (2011)
17. Simon, P., Taylor, M., Kiss, I.: Exact epidemic models on graphs using graph-automorphism driven lumping. *J. Math. Biol.* **62**, 479–508 (2011)
18. Tudisco, F., Arrigo, F., Gautier, A.: Node and layer eigenvector centralities for multiplex networks. *SIAM J. Appl. Math.* **78**(2), 853–876 (2018)
19. Van Mieghem, P., Omic, J., Kooij, R.: Virus spread in networks. *IEEE/ACM Trans. Netw.* **17**(1), 1–14 (2009)
20. Wei, X., Valler, N., Aditya Prakash, B., Neamtiu, I., Faloutsos, M., Faloutsos, C.: Competing memes propagation on networks: a case study of composite networks. *ACM SIGCOMM Comput. Commun. Rev.* **42**, 5–11 (2012)



Microscopic Markov Chain Approach for Measuring Mobility Driven SARS-CoV-2 Transmission

Trevor G. Kent¹⁽⁾, Nolan E. Phillips¹, Ian McCulloh^{1,2}, Viveca Pavon-Harr¹, and Heather G. Patsolic¹

¹ Applied Intelligence Accenture Federal Services, 1201 New York Ave NW,
Washington, DC 20005, USA

trevor.g.kent@accenturefederal.com

² Whiting School of Engineering, Johns Hopkins University, Laurel, MD, USA

Abstract. After more than a year of non-pharmaceutical interventions, such as, lock-downs and masks, questions remain on how effective these interventions were and could have been. The vast differences in the enforcement of and adherence to policies adds complexity to a problem already surrounded with significant uncertainty. This necessitates a model of disease transmission that can account for these spatial differences in interventions and compliance. In order to measure and predict the spread of disease under various intervention scenarios, we propose a Microscopic Markov Chain Approach (MMCA) in which spatial units each follow their own Markov process for the state of disease but are also connected through an underlying mobility matrix. Cuebiq, an offline intelligence and measurement company, provides aggregated, anonymized cell-phone mobility data which reveal how population behaviors have evolved over the course of the pandemic. These data are leveraged to infer mobility patterns across regions and contact patterns within those regions. The data enables the estimation of a baseline for how the pandemic spread under the true ground conditions, so that we can analyze how different shifts in mobility affect the spread of the disease. We demonstrate the efficacy of the model through a case study of spring break and its impact on how the infection spread in Florida during the spring of 2020, at the onset of the pandemic.

Keywords: Microscopic Markov Chain Approach · Approximate Bayesian Computation · Epidemiology · SARS-CoV-2

1 Introduction

During the bubonic plague outbreak of 1665–1666, the town of Eyam stands out as an example of how limiting mobility can alter the spread of a deadly disease. After the bubonic plague came to Eyam through trade with London, the town and its neighboring communities developed a plan that would enforce

the first example of 'curbside pick-up' by shutting down their borders to flow in and out of the community while neighboring communities dropped off much needed supplies outside those borders [4, 27]. As author and prominent YouTube content creator John Green points out in a podcast episode of the Anthropocene Reviewed [14], "Eyam's self-quarantine held to a remarkable extent for fourteen months, preventing the spread of the mortality to any nearby towns." While fascinating, the convolutions of human connection did not exist to the extent that it does today. We live in an increasingly interconnected society where technological advancements make travel across vast distances more accessible. The United Nations World Tourism Organization estimates there were nearly 1.4 billion international arrivals in the year 2018; a figure that has more than doubled since the year 2000 [24]. With this increased travel comes more wide-spread transmissibility of communicable diseases which are no longer confined to the locations from which they originate. The SARS-CoV-2 pandemic has particularly illuminated just how connected our world has become, and the effect of human connectivity on disease transmission.

The advent of widespread use of mobile phones and GPS devices has created an unprecedented amount of data which captures the locations of individuals at varying time-points. With more granular details about human mobility patterns, researchers have been able to leverage these data to help explain and predict the spread of disease while developing epidemiological models and algorithms. Researchers in [2] were able to use mobility and social distancing parameters to model spread of COVID-19 during the early stages of the pandemic in Spain. However, due to their limited access to mobility data, they were forced to adopt a rather complex model that introduced dynamic contact parameters by applying deviations from their "baseline" conditions. Our approach differs in that we adopt time varying mobility and contact parameters derived from Cuebiq that mitigates uncertainty arising from variation of human behavior before, during, and after lockdown interventions. We show that access to these data allows us to achieve a similar fit to observed daily cases while maintaining a parsimonious functional form for the probability of infection. Researchers in [5] also adopt a probability of transmission with dependencies on human mobility and contact parameters derived from mobile phones. They were able to show how model fits improved given more realistic mobility networks for 10 major commuting regions in the United States. While this method shows promise, limiting the scope of analysis to a commuting region imposes artificial boundaries on the system. We apply a novel approach that includes an extra node in the mobility matrix which encodes the rest of the U.S. and thereby permits infections initialized elsewhere to spread into the region of interest.

As was found in [20], it is crucial that we examine more granular transmission patterns of a disease to better understand its spread. While the authors of [20] focus on vaccination, which is not relevant for the time frame of this study, looking at how people are mobilizing and coming into contact with one another at a more granular scale than country or state level helps us better understand how things like adherence to stay-at-home protocols and across-region mobility

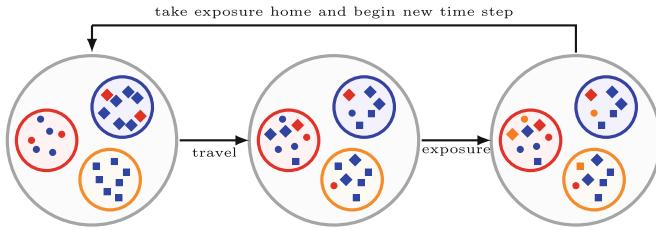


Fig. 1. Considering a population which has three sub-populations, we provide an illustrative example of how susceptible (blue) and infected (red) individuals mobilize and interact during a single time step. Individuals who transition to being exposed during this illustrative time-step are shown in (orange).

impact the spread of a disease. Studies, such as the one presented in [11], have argued that stay-at-home orders have lead to reduced across-region mobility, mitigating the spread of disease. Cuebiq's data allow us to include mobility and contact trends at the county level, for a more refined model.

In this manuscript, we introduce a Microscopic Markov Chain Approach (MMCA) SEIRD compartmental model that incorporates mobility and contact patterns, building upon the work in [2, 5]. Using this model, we demonstrate how these data influence the spread of a contagious disease. Our model aims to better account for the effects of mobility on the likelihood of becoming infected. We assume each individual starts in a particular spatial unit; hereafter, referred to as patch. At each time step, they have the ability to move to a different patch, interact with individuals in that patch, and then return to their initial patch. Thus, the probability of a susceptible person becoming infected at a given time point depends on the patch where interactions occur and the demographics of the population in that patch at that time. Figure 1 illustrates an example of how individuals might move around at a given time point in a population with 3 patches. We see three states of infection (red, blue, orange) where blue dots correspond to susceptible individuals and red dots correspond to infected individuals. Reading the figure from left to right, we see that during a time-step individuals start in their patch, travel to other patches where they mingle, and then transition to being exposed (orange) before returning to their original patch for the next time-step. For ease of illustration, we do not include nodes for individuals in the R/D compartments.

The article proceeds as follows. First we will delineate the technical details of the model in Sect. 2. In Sect. 3 we provide a thorough analysis of the spread of COVID-19 in the state of Florida at the county level where we explain the data estimates Sect. 3.1, fitting procedure Sect. 3.2, and scenario analyses Sect. 3.3 explored in this study. Finally, we conclude with a discussion Sect. 4 on findings and explore other potential avenues of research.

2 Technical Details

We have a population of n individuals who are partitioned into patches $\{1, \dots, N\}$. We will use the notation $\rho_i^m(t)$ to denote the proportion of individuals in patch $i \in \{1, \dots, N\}$ and state $m \in \{S, E, I, R, D\}$ at time t , where time is measured in days. We adopt a Markov model in which tomorrow's state of disease is only dependent on the current state.

$$\rho_i^S(t+1) = (1 - \Pi_i(t))\rho_i^S(t) \quad (1)$$

$$\rho_i^E(t+1) = \Pi_i(t)\rho_i^S(t) + (1 - \sigma)\rho_i^E(t) \quad (2)$$

$$\rho_i^I(t+1) = \sigma\rho_i^E(t) + (1 - \gamma(1 - \omega) - \omega\delta)\rho_i^I(t) \quad (3)$$

$$\rho_i^R(t+1) = (1 - \omega)\gamma\rho_i^I(t) \quad (4)$$

$$\rho_i^D(t+1) = \omega\delta\rho_i^I(t) \quad (5)$$

We assume static parameters for transitions from $E \rightarrow I$, $I \rightarrow R$, and $I \rightarrow D$. However, due to mobility and social distancing, the primary subject of this study $\Pi_i(t)$, which represents the transition probability from state S at time t to state E at time $t+1$, varies with both time and space. The probability of a randomly selected individual in patch i making such a transition can be expressed as the sum of the probabilities of getting infected in each of the patches multiplied by the probability the individual went to a different patch (Bayes' Rule). Let the matrix $R(t)$ denote the mobility matrix where $R_{ij}(t)$ denotes the proportion of individuals in patch i who mobilize to patch j at time t , and let $P_j(t)$ denote the probability of being infected while in patch j at time t . Note that $\sum_{j=1}^N R_{ij}(t) = 1$ for all $i \in \{1, \dots, N\}$, and all time-steps t . We can express $\Pi_i(t)$ as the convex combination of the proportion of individuals in patch i getting infected by the contagion in patch j , over all j , at time t .

This can be expressed as follows:

$$\Pi_i(t) = \sum_{j=1}^N R_{ij}(t)P_j(t), \quad (6)$$

where $P_j(t)$ captures the probability that a person becomes infected by the contagion while in patch j . This, again, depends on mobility, since the probability a person becomes infected while in patch j depends on how people are mobilizing into and out of that patch, how many other individuals a susceptible individual can expect to come into contact with while in patch j , and what proportion of those individuals will be carriers of the disease.

Let $X(t; j \rightarrow i)$ denote the number of susceptible individuals who are exposed by an infected individual who came to patch i from patch j at time t . Further assume exposure is independent across individuals. We can model $X(t; j \rightarrow i)$ as a Binomial random variable with parameters $Q(t; j \rightarrow i)$ and β where $Q(t; j \rightarrow i)$ represents the number of contacts between individuals in patch i with individuals coming into patch i from patch j at time t and β represents the rate of transmission, which is assumed to be fixed over time and patches.

$$\begin{aligned} P_j(t) &= 1 - \prod_{k=1}^N P(X(t; k \rightarrow j) = 0) \\ &= 1 - \prod_{k=1}^N (1 - \beta)^{Q(t; k \rightarrow j)}. \end{aligned}$$

To compute $Q(t; k \rightarrow j)$, which is the proportion of infected individuals coming into patch j from patch k at time t scaled by the expected number of contacts, we need to multiply the average number of contacts for someone in patch j at time t , $c_j(t)$, by the proportion of *infected* individuals in patch j who mobilized from patch k at time t . Thus, we can express $Q(t; k \rightarrow j)$ as follows:

$$Q(t; k \rightarrow j) = c_j(t) \frac{n_{k \rightarrow j}^I(t)}{n_j^{eff}(t)}, \quad (7)$$

where

$$n_{k \rightarrow j}^I(t) = \rho_k^I(t) n_k R_{kj}(t) \eta_k(t)$$

represents the proportion of infected individuals from patch k mobilizing to patch j at time t . Here $\eta_k(t)$ denotes the proportion of infectious individuals in patch k who choose to mix with the general population,

$$\eta_k(t) = (1 - \alpha_k(t))[\lambda_0 - \lambda_c \mathcal{H}(t - t_c)] \quad (8)$$

where,

$$\mathcal{H}(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

and finally,

$$n_j^{eff}(t) = \sum_{k=1}^N n_k R_{kj}(t)$$

represents the effective population of patch j at time t , that is, the total number of individuals in patch j at time t taking into consideration individuals staying in the patch and coming in from other patches.

Taken together, the probability of becoming infected by the contagion while in patch j is

$$P_j(t) = 1 - \prod_{k=1}^N (1 - \beta)^{Q(t; k \rightarrow j)} \quad (9)$$

$$= 1 - (1 - \beta)^{\frac{c_j(t) \sum_{k=1}^N n_{k \rightarrow j}^I(t)}{n_j^{eff}(t)}}. \quad (10)$$

Thus, $\Pi_i(t)$ can be computed as follows.

$$\Pi_i(t) = \sum_{j=1}^N R_{ij}(t) \left(1 - (1 - \beta)^{\frac{c_j(t) \sum_{k=1}^N n_{k \rightarrow j}^I(t)}{n_j^{eff}(t)}} \right). \quad (11)$$

The above formulation does not take into consideration those individuals who are staying home, and assumes that people who are infected mobilize outside the resident patch at a rate proportional to the population. We incorporate the proportion of individuals staying home with the individuals remaining in their respective patches at time t . Allow $Z(t)$ to be a matrix such that row i represents the distribution of mobility of individuals across patches out of those who are mobilizing and are from patch i , while row i of $R(t)$ represents the distribution of individuals across patches out of those individuals who begin time t being in patch i . This means that $R_{ij}(t) = \kappa_i(t)(I_N)_{ij} + (1 - \kappa_i(t))Z_{ij}(t)$ where $\kappa_i(t)$ denotes the proportion of individuals staying home at time t and $(I_N)_{ij}$ represents an element of the identity matrix with $N \times N$ dimensions. Then $Z_{ii}(t)(1 - \kappa_i(t))$ denotes the proportion of individuals who are not staying home at time t but are mobilizing *within* patch i (Fig. 2).

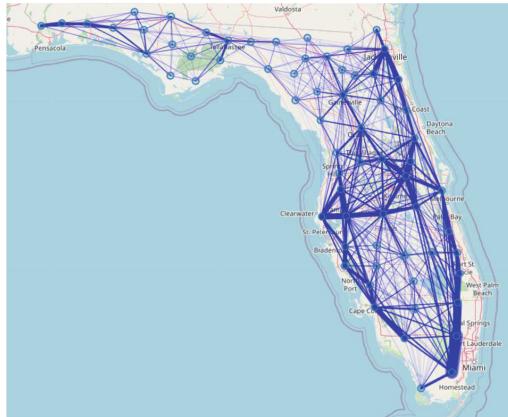


Fig. 2. The probability of infection updates each day according to a Markov Process and patches are linked through an underlying mobility matrix $R(t)$. Displayed here using scikit-mobility [23] is the mobility network for a snapshot in time where the thickness of the connections are motivated by the proportion of people from the source county that travel to a target county. Note that the extra node in the mobility matrix to denote the rest of the U.S. is not shown here.

In practice it is important to note that the region under consideration is not a closed system. The boundary issue is a long-established problem within network and spatial analyses [18, 28]. Rather than imposing an artificial boundary that would explicitly assume infections cannot be imported from areas outside the focal area, we include a “catch all” node to incorporate mobility patterns from all other areal units. The mobility and contact parameters are averaged over all exogenous patches. Though this is still an abstraction from adopting a model over the region at large, it is an improvement over approaches that impose boundaries around the area of study. In our case study, introduced in Sect. 3, we show how

the MMCA SEIRD model can be applied to the spread of SARS-CoV-2 in Florida and add an extra node to encapsulate mobility between Florida and the rest of the U.S.

3 Florida and Mobility During the Pandemic

We demonstrate the utility of our model through a case study of SARS-CoV-2 spread from January through May of 2020 at the county level in Florida. We focus on the months of March and April, surrounding spring break where we observe larger volumes of people in Florida than normal.

3.1 Data and Parameter Estimates

The majority of parameters in the model are either estimated directly from Cuebiq [1] or pulled from the literature. To estimate the unknown parameters, we employ an Approximate Bayesian Computation (ABC) algorithm as defined in [8]. This family of algorithms is advantageous when the likelihood function is expensive or intractable to calculate, which is the case here since we would otherwise have to sum over all latent paths from which infections could have arisen. The unknown parameters of our model consist of the transmission rate β , initial number of latent infectious seeds e_0 , initial number of infectious seeds i_0 , proportion of infectious individuals who choose to mix with the general population despite being infectious λ_0 and then following the onset of lockdown restrictions this figure is reduced by λ_c . The method yields means for the posterior distributions of each variable.

Parameter	Value	Description	Source
$c_j(t)$	–	Average contacts	[1]
$R_{ij}(t)$	–	Mobility matrices	[1]
$\kappa_i(t)$	–	Proportion staying home	[1]
$\alpha_k(t)$	–	Social distancing index	[29]
$p_i(t)$	–	Prevalence ratio	[15]
ζ	14	Reporting lag	[15]
σ	1/5.1	Incubation rate	[17]
γ	1/21	Recovery rate	[26]
δ	1/17.8	Death rate	[26]
ω	0.0066	Infection fatality ratio	[26]
β	0.0649	Transmission rate	Calibrated
e_0	24.988	Initial pre-infectious seeds	Calibrated
i_0	9.849	Initial infectious seeds	Calibrated
λ_0	0.8991	Infectious non-compliance pre-lockdown	Calibrated
λ_c	0.2499	Non-compliance adjustment post-lockdown	Calibrated

The U.S. department of health and human services reports data on important intervention dates like containment or reopening events at the state and county level [7]. They report that many counties in Florida didn't officially shutdown until late March or early April. Thus we chose to impose our lockdown restriction t_c at March 30, 2020. The first official case in the United States was reported on January 21, 2020 [10], yet there is still uncertainty surrounding when the first infections appeared and how many. We chose to start the pandemic on December 30, 2019 (the day before the first confirmed case reported in China [10]). We estimate the initial infectious seeds i_0 as 10 (10 times the number known in China), and initial exposed cases e_0 as 25 in the region outside Florida. The ratio of initial exposed to initial infectious seeds falls within the credible range of what R_0 for SARS-CoV-2 is estimated to be [21,30] (Fig. 3).

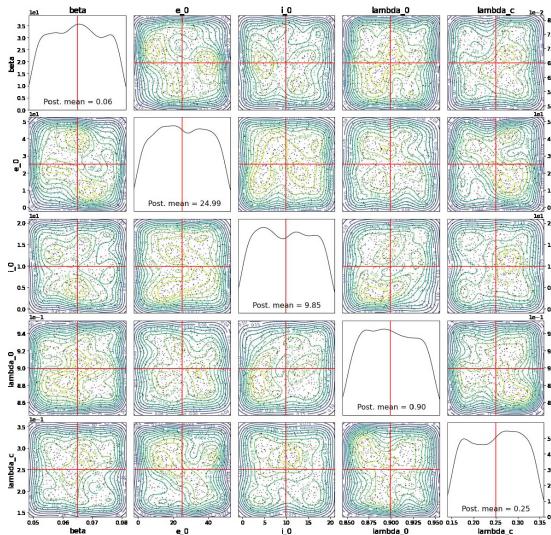


Fig. 3. The posterior distributions of the unknown parameters in the model. We use the Python package abcpy [9] with 1000 samples.

The University of Maryland reports a data series $\alpha_k(t)$ that measures the extent to which social distancing restrictions are being followed at the county level [29]. We use this parameter in Eq. 8 along with λ_0 and λ_c to measure the proportion of infectious individuals who are not adhering to social distancing restrictions before and after lockdown at time t_c . Before the pandemic, the United States fostered a culture that encouraged individuals to go to work whenever possible, including when sick. A prior CDC study found that roughly 82.7% of individuals went to work even though they were exhibiting flu-like symptoms [12]. We surmise that during the early months of the pandemic prior to lockdown restrictions, people who were infectious likely went about their business

as usual, only staying home when they felt they could afford to take time off. Once citizens became more aware that the disease was a more dangerous threat, lock-downs started being mandated and policies enacted to ensure people would get tested and stay home while sick.

3.2 Residuals: True Estimates and the Markov Model

Due to the asymptomatic nature of the disease [19] and limited testing availability early on in the pandemic [25], a significant quantity of cases went undetected [22, 25]. These missing observations present challenges for estimating unknown parameters in an epidemiological model. Several researchers have found success in fitting to COVID-19 attributed death counts instead [2, 22]. The authors in [22] implement a Bayesian augmentation algorithm and were able to validate the model with sero-prevalence data from New York state. However, this approach cannot be applied to the earliest stages of the pandemic at more granular spatial areas since not enough deaths had been reported. An independent researcher Youyang Gu provides an alternative approach reliant on testing data, which became one of the most widely-cited models in academia, media, government, and industry [15]. Along with his infections estimates, he provides a prevalence ratio, which we will denote as $p_i(t)$, and represents the ratio of true new daily infections to daily reported cases. We choose to fit to reported new daily cases after correcting for an assumed $\zeta = 14$ day lag in reporting and applying the prevalence ratio. Thus to arrive at our estimate of new daily cases we first calculate infectious incidence $N_i^{E \rightarrow I}(t)$ like so:

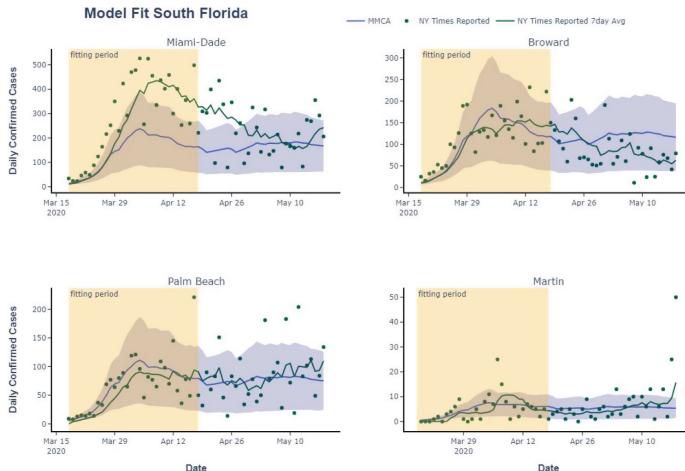


Fig. 4. For South Florida, we show reported new daily cases from *New York Times* (green) vs new daily cases estimated from our MMCA SEIRD model (blue). We choose to fit the model from 03-18-2020 to 04-18-2020. The shaded region represents the 95% confidence interval for our estimates.

$$N_i^{E \rightarrow I}(t) = \sigma \rho_i^E(t) n_i \quad (12)$$

The estimated daily cases is then given by:

$$\bar{N}_i^{E \rightarrow I}(t) = \frac{N_i^{E \rightarrow I}(t - \zeta)}{p_i(t)} \quad (13)$$

In Fig. 4 we show how our estimated daily cases $\bar{N}_i^{E \rightarrow I}(t)$ compares to the reported 7 day average of new daily cases from the *New York Times*. We demonstrate that under baseline population behaviour derived from Cuebiq, the model is able to fit well within the 1 month fitting period from March 18th to April 18th, as well as a 1 month held out period from April 18th to May 18th. We first apply our model to data from March 18th since it is the earliest date that we have prevalence ratio data for all counties in South FL.

3.3 Alternate Realities

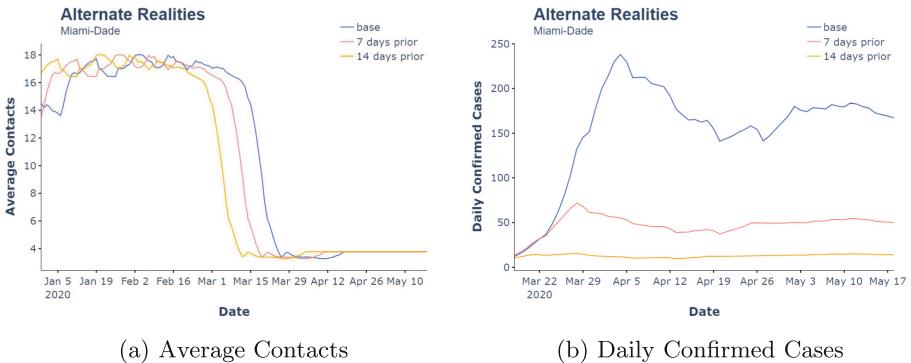


Fig. 5. Displayed here are model estimates of average contacts $c_j(t)$ (a) and new daily reported cases $\bar{N}_i^{E \rightarrow I}(t)$ (b) given two alternative scenarios. The scenario where lock down occurs 7 days prior to t_c is shown in (salmon) while the scenario where lockdown occurs 14 days prior to t_c is shown in (orange). The scenario given real data inputs is shown in (blue). Note that we replace the true input parameters with the last known input parameters for all time points after the last day of the fitting period.

Once the default parameters are set, a user can investigate the impact of alternate realities, such as, never shutting down or allowing lockdown restrictions to persist over a much longer period. Here, we examine the scenarios where lockdown restrictions occur 7 days and 14 days prior to the official enforcement dates. All data derived input parameters are lagged to simulate the effect of the shift in human behaviour as a result of awareness campaigns or enforcement of containment policies occurring earlier. In Fig. 5 we see an example of how $c_j(t)$

was adjusted for each scenario. We also impose a shifted lockdown date t_c so that the fitted λ_c parameter is applied earlier. This approach inherently assumes that the effect of lockdown restrictions would be the exact same when applied earlier in the pandemic.

Marginal changes in intervention dates can lead to vastly different outcomes in disease transmission. In Fig. 5, we see that our model suggests applying restrictions even a week earlier would have reduced the daily reported cases by a wide margin. On April 4, the first peak in our model, we find under the condition where lockdown occurs 7 days prior that the daily case count would reduce by 182 cases or 76.8% change. Over the entire two month period, the total number of cases would reduce by 6633 (69.2%) under the same alternative scenario. Locking down prior to spring break could have not only prevented the influx of a large volume of people into Florida, but could also likely have had a huge impact on how the pandemic played out in the rest of the U.S. due to a subset of the influx of people returning back to their resident counties with the disease. More careful selection of when lockdown restrictions are applied could have drastically reduced the strain on our public health system.

4 Discussion

We have shown that behavioral changes drive the differences in COVID-19 transmission. We use the lock-down dates as a proxy for when people change their behavior most; however, these changes are likely dynamic and heavily influenced by increased awareness and adherence to local guidelines. We do find evidence that the shift in behavior that occurred around March 30, 2020 was influential in determining how the pandemic spread in Florida. It is unclear whether the change of behavior we observe is a direct result of the lock-down restrictions themselves, awareness of the disease, or some other factor. Research is beginning to emerge on the co-evolution of awareness and epidemic progression; see, for example, [13] in which the authors adopt a multiplex network and MMCA approach. While assessing the effects of lock-downs and other formal policies on human behavior is beyond the scope of this paper, future research should seek to account for public awareness of a disease and its effect on human interactions over the course of a pandemic within this modeling framework.

There is significant evidence that a large portion of the population experienced infectiousness without showing symptoms. In fact, one study reported that 39.7 and 44.8% of positive cases sampled for two different survey periods in the Italian municipality of Vo' were found to be asymptomatic [19]. Parameter estimation can be computationally intensive, which constrains how much realism can be captured within the functional form. Though we did not include an asymptomatic compartment, we did address the fact that asymptomatic individuals are likely to mobilize differently than their symptomatic counterparts. We argue that symptomatic individuals are more aware of their illness and thus more likely to isolate themselves as a result. We regulate the extent to which discordant pairs can be formed by incorporating the stay home parameter into

the calculation of the mobility matrices. Equation 8 also captures the fact that a subset of the population will continue to mix with the population despite being infectious. This parameter plays an important role in capturing how differently asymptomatic individuals may respond to infectiousness. However, this approach does not account for the fact that the infectiousness profile may be different for asymptomatic vs symptomatic patients, and existing research proves that transmission events can occur 2 to 3 days before symptom onset [16]. Future research should seek to add this additional complexity to the model by introducing additional compartments for asymptomatic and pre-symptomatic patients.

Our model is notably less accurate at quantifying the number of cases for Miami-Dade than the other counties in Florida early in the pandemic. There is another layer of human mobility that is not captured in the commuting flows; namely airline traffic. Miami, an international hub, should have increased numbers of people and contacts, thus infections, imported from outside of the commuting region. This is an established problem in epidemiological modeling and prior research has shown the importance of accounting for airline travel as another vector of transmission [3, 6]. With airline data, our approach could be extended to account for international travelers and even different rates of positivity of those travelers.

Furthermore, the exposure and transmission rates change with the introduction of vaccines and variants over the course of a pandemic. Our approach could be adjusted to account for this heterogeneity by dividing the pandemic into epochs. The end conditions of one epoch become the seed conditions for the next and the unknown parameters associated with the primary drivers of disease during that period of the pandemic could be estimated. This would require creating additional exposed and infected compartments for the variants. Similarly, an additional compartment could be created for vaccinated individuals since “break through” cases have been identified. Thus, more of the nuances of the current (or future) pandemics could be accounted for using this modeling paradigm. Though it should be noted that increased complexity will make the model take longer to run and could introduce additional errors when estimating unknown parameters.

This work enhances our ability to model future communicable diseases and the potential effects of non-pharmaceutical interventions. The approach accounts for many of the drivers of the SARS-CoV-2 pandemic, yet remains adaptable to emerging conditions. Thus, this framework empowers policy and decision makers with data-driven estimates of how the pandemic will affect specific regions and enable those decision makers to enact more efficacious policies.

Acknowledgements. This model was built in collaboration with Applied Intelligence Accenture Federal Services. The past year we served as members of their COVID-19 response team where we built and deployed models for decision support of various federal clients. We also appreciate Cuebiq’s Data for Good program for providing the mobility data needed for this analysis.

References

1. Cuebiq. <https://help.cuebiq.com/hc/en-us>
2. Arenas, A., et al.: Modeling the spatiotemporal epidemic spreading of COVID-19 and the impact of mobility and social distancing interventions. *Phys. Rev. X* **10**(4), 041055 (2020)
3. Bajardi, P., Poletto, C., Ramasco, J.J., Tizzoni, M., Colizza, V., Vespignani, A.: Human mobility networks, travel restrictions, and the global spread of 2009 H1N1 pandemic. *PloS one* **6**(1), e16591 (2011)
4. Beaumont, P.: Eyam recalls lessons from 1665 battlewith plague. *The Observer* (2020)
5. Chang, S., et al.: Mobility network models of COVID-19 explain inequities and inform reopening. *Nature* **589**(7840), 82–87 (2021)
6. Colizza, V., Barrat, A., Barthélémy, M., Vespignani, A.: The role of the airline transportation network in the prediction and predictability of global epidemics. *Proc. Natl. Acad. Sci.* **103**(7), 2015–2020 (2006)
7. COVID-19 State and County Policy Orders. [Dataset]. U.S. Department of Health and Human Services (2021). https://catalog.data.gov/sr_Latn/dataset/covid-19-state-and-county-policy-orders-9408a
8. Del Moral, P., Doucet, A., Jasra, A.: An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Stat. Comput.* **22**(5), 1009–1020 (2012)
9. Dutta, R., Schoengens, M., Onnela, J.-P., Mira, A.: ABCPy: a user-friendly, extensible, and parallel library for approximate Bayesian computation. In *Proceedings of the Platform for Advanced Scientific Computing Conference*, pp. 1–9 (2017)
10. Editors, H.: First confirmed case of COVID-19 found in U.S. (2020). <https://www.history.com>this-day-in-history/first-confirmed-case-of-coronavirus-found-in-us-washington-state>
11. Engle, S., Stromme, J., Zhou, A.: Staying at home: The mobility effects of COVID-19. VOX, CEPR Policy Portal (2020). <https://voxeu.org/article/staying-home-mobility-effects-covid-19>
12. Cdc, P., et al.: Experiences with influenza-like illness and attitudes regarding influenza prevention-united states for disease control, 2003–04 influenza season. MMWR. Morbidity Mortality Weekly Rep. **53**(49), 1156–1158 (2004). <https://www.cdc.gov/mmwr/preview/mmwrhtml/mm5349a3.htm>
13. Granell, C., Gómez, S., Arenas, A.: Competing spreading processes on multiplex networks: awareness and epidemics. *Phys. Rev. E* **90**(1), 012808 (2014)
14. Green, J.: Plague. [podcast] The Anthropocene Reviewed (2020). <https://www.wnycstudios.org/podcasts/anthropocene-reviewed/episodes/anthropocene-reviewed-john-green-plague>, Accessed 5 Oct 2021
15. Gu, Y.: COVID-19 Projections for Miami-Dade, Fl, USA (2020). <https://covid19-projections.com/us-fl-miami-dade>
16. He, X., et al.: Temporal dynamics in viral shedding and transmissibility of COVID-19. *Nat. Med.* **26**(5), 672–675 (2020)
17. Lauer, S.A., et al.: The incubation period of coronavirus disease 2019 (COVID-19) from publicly reported confirmed cases: estimation and application. *Ann. Internal Med.* **172**(9), 577–582 (2020)
18. Laumann, E.O., Marsden, P.V., Prensky, D.: The boundary specification problem in network analysis. *Res. Methods Soc. Netw. Anal.* **61**, 87 (1989)
19. Lavezzo, E., et al.: Suppression of a SARS-CoV-2 outbreak in the Italian municipality of Vo'. *Nature* **584**(7821), 425–429 (2020)

20. Masters, N.B., Eisenberg, M.C., Delamater, P.L., Kay, M., Boulton, M.L., Zelner, J.: Fine-scale spatial clustering of measles non vaccination that increases outbreak potential is obscured by aggregated reporting data. *Proc. Natl. Acad. Sci.* **117**(45), 28506–28514 (2020)
21. Li, Q., et al.: Early transmission dynamics in Wuhan, China, of novel coronavirus-infected pneumonia. *New Engl. J. Med.* **382**, 1199–1207 (2020)
22. McCulloh, I., Kiernan, K., Kent, T.: Inferring true COVID19 infection rates from deaths. *Front. Big Data* **3**, 37 (2020)
23. Pappalardo, L., Simini, F., Barlacchi, G., Pellungrini, R.: scikit-mobility: a Python library for the analysis, generation and risk assessment of mobility data (2019)
24. Roser, M.: Tourism. Our World in Data (2017). <https://ourworldindata.org/tourism>
25. Silverman, J.D., Hupert, N., Washburne, A.D.: Using influenza surveillance networks to estimate state-specific prevalence of SARS-CoV-2 in the United States. *Sci. Transl. Med.* **12**(554), 1–9 (2020)
26. Verity, R., et al.: Estimates of the severity of coronavirus disease 2019: a model-based analysis. *Lancet Infect. Dis.* **20**(6), 669–677 (2020)
27. Whittles, L.K., Didelot, X.: Epidemiological analysis of the Eyam plague outbreak of 1665–1666. *Proc. Royal Soc. B Biol. Sci.* **283**(1830), 20160618 (2016)
28. Wang, Q., Phillips, N.E., Small, M.L., Sampson, R.J.: Urban mobility and neighborhood isolation in America’s 50 largest cities. *Proc. Natl. Acad. Sci.* **115**(30), 7735–40 (2018)
29. Zhang, L., et al.: An interactive COVID-19 mobility impact and social distancing analysis platform. *medRxiv* (2020)
30. Zhao, S., et al.: Preliminary estimation of the basic reproduction number of novel coronavirus (2019-nCoV) in China, from 2019 to 2020: A data-driven analysis in the early phase of the outbreak. *Int. J. Infect. Dis.* **92**, 214–217 (2020). PMID: 32007643; PMCID: PMC7110798. <https://doi.org/10.1016/j.ijid.2020.01.050>



Epidemics in a Synthetic Urban Population with Multiple Levels of Mixing

Alessandro Celestini¹, Francesca Colaiori^{2,3}, Stefano Guarino^{1(✉)}, Enrico Mastrostefano¹, and Lena Rebecca Zastrow¹

¹ Institute for Applied Mathematics “Mauro Picone”, CNR, Rome, Italy

² Institute for Complex Systems, CNR, Rome, Italy

³ Department of Physics, Sapienza University, Rome, Italy

Abstract. Network-based epidemic models that account for heterogeneous contact patterns are extensively used to predict and control the diffusion of infectious diseases. We use census and survey data to reconstruct a geo-referenced and age-stratified synthetic urban population connected by stable social relations. We consider two kinds of interactions, distinguishing daily (household) contacts from other frequent contacts. Moreover, we allow any couple of individuals to have rare fortuitous interactions. We simulate the epidemic diffusion on a synthetic urban network for a typical medium-sized Italian city and characterize the outbreak speed, pervasiveness, and predictability in terms of the socio-demographic and geographic features of the host population. Introducing age-structured contact patterns results in faster and more pervasive outbreaks, while assuming that the interaction frequency decays with distance has only negligible effects. Preliminary evidence shows the existence of patterns of hierarchical spatial diffusion in urban areas, with two regimes for epidemic spread in low- and high-density regions.

Keywords: SIR · Epidemic · Social network · Data driven · Urban system

1 Introduction

The prediction and control of infectious diseases have an enormous impact on public health, the economy, and society. Mathematical modeling and computer simulations provide powerful tools to understand the dynamics of epidemic outbreaks and design strategies to control and mitigate them. The modeling of epidemic contagion has evolved from simple compartmental schemes to sophisticated data-informed models using synthetic populations whose demographics are statistically indistinguishable from the census data. As in other fields, this evolution towards more realistic models involves a trade-off between simplicity and accuracy of the predictions.

The classical compartmental models of disease propagation [1,2] describe the system at the population-level by grouping individuals in sub-populations according to their health states relevant for transmission and tracking changes in the sizes of the sub-populations, without specifying which individuals are involved. The movements in and out of the compartments are governed by constant transition rates in ordinary differential equations, corresponding to exponentially distributed waiting times in the compartments. This approach assumes a fully-mixed population where an infective individual is equally likely to spread the disease to any other individual, and it is equivalent to a mean-field.

More recent works focused on incorporating heterogeneous contact patterns in the models. In individual-based models, the population-level behavior emerges from the microscopic interactions between agents that can carry a set of attributes such as age, spatial location, and social behavior. Socio-demographic attributes collected from census data and surveys, or integrated with mobile, traffic or wearable sensor data and online tools [3–5], allow the construction of realistic contact matrices describing the mixing patterns in typical social settings (e.g., households, schools, workplaces) [3,6–8]. Meta-population models built using data from inter-population mobility [8–11] have permitted remarkable progress in describing how the disease travels from one city/nation to the other and estimating epidemic paths predictability [12]. Compartmental models defined with stochastic and time-varying disease transmission rates [13] can include changes in the transmission rate that might occur due to the implementation of control measures and variations due to unobserved processes.

By representing individuals in the population as vertices of a graph, network-based models aim to understand how the topological structure of the contact network affects the spreading dynamics upon them [14–18]. Starting from actual epidemiological data, a recent body of research work [19–21] tried to reconstruct the relationship between population heterogeneity and the spread of infectious diseases in a population or territory. Due to the lack of granular disease data, a clear picture is still missing. The influence of population density and distances, examined at different scales (world vs. states vs. cities), leads to different results on the main driver of diffusion.

Ball and Neal [22,23] addressed the need to incorporate randomness in network-based epidemic models studying a stochastic Susceptible–Infected–Removed (SIR) model in a closed population where individuals make *local* contacts on a social network and *global* contacts with individuals chosen at random in the population. This model is also suited to mimic multiple routes of transmission [24].

In this paper, we address similar questions for epidemics at the urban scale by simulating an agent-based SIR model in a synthetic urban population. As in [23], our model incorporates casual contacts on top of a structured set of social relations. However, we use a data-informed model to obtain an age-structured population connected by a geographically referenced social network, and we allow for global casual contacts to be dependent on distance and/or on the age class of an agent. By encoding in our epidemic model the dependence of contact frequencies on age and distance, we can follow the geographical and age-stratified evolution of the outbreak under different configurations, without the need for high-resolution data about transportation, mobility, workplaces, or schools –

which, however, may be incorporated in the model in the future. The goal is to understand how, in a realistic population bound by a data-driven social network, the combination of different levels of mixing and their possible dependence on socio-demographic and geographic features of the host population impact the epidemic diffusion process.

2 Models, Materials and Methods

We consider the *territory* of the Municipality of Viterbo, Italy, represented as a rectangular bounding box, partitioned into a grid of T square tiles of fixed side $l = 500$ m. We build a *synthetic population* of $N \approx 60K$ geo-referenced and age-stratified agents using density estimates provided by the WorldPop Project [25] and keep only tiles having at least 10 residents. Each agent is assigned to an age group based on census data aggregated at the provincial level. We consider four age-groups: *children* (0 to 17), *young* people (18 to 34), *adults* (35 to 64), and *elderly* (65+). The N agents form the vertex set V of an urban social network (USN), i.e., an unweighted undirected graph $G = (V, E)$ obtained as the union of the *household graph* $G_H = (V, E_H)$ and *acquaintance graph* $G_A = (V, E_A)$, where $E = E_H \cup E_A$ and $E_H \cap E_A = \emptyset$. The graph G_H consists of a set of *cliques*, one per household, inferred from census data under the assumption that: (i) all members of a household live in the same tile; (ii) children are younger than their parents; (iii) partners have, on average, a similar age. The algorithm used to group individuals into households is described in detail in [26]. The resulting household graph has a data-driven average degree ν , which has been found empirically to be ≈ 2 . To construct the acquaintance graph G_A , for each pair (u, v) , the edge (u, v) is added to E_A independently of all others with probability $\psi_{u,v} = \Pr[(u, v) \in E_A] \propto s_{g_u, g_v} \cdot d(u, v)^{-1} \cdot f_u \cdot f_v$, where: $s_{i,j}$ is the age-based social mixing for groups i and j , deduced from aggregated contact data [3] through the SOCRATES Data Tool [27]; $d(u, v) = \max\{\frac{l}{2}, d^*(u, v)\}$ is the discretized distance between u and v , where $d^*(u, v)$ is the distance between the centers of u and v 's tiles of residence; f_u is the vertex-intrinsic social fitness [28] of vertex u , which controls u 's sociability, i.e., tendency to make friends. The choice of 1 as the distance dependence exponent, and of $f_u \sim 1 + \text{Lognormal}(\ln(2), \frac{1}{4})$ as used in the following is justified on an empirical basis in [26] to best represent social contacts at the urban level. A generalized formulation of the USN model used here is given in [29] and [26], where it is also shown that the main topological properties, s.a. the degree distribution, the clustering coefficient, and the average distance, are preserved in different network realizations. The implementation of the considered USN model is freely available as open-source software at gitlab.com/cranic-group/usn.

We simulate the disease propagation using an individual-based continuous-time SIR model: at each time step, each infected individual u spreads the disease to each susceptible individual v with rate $p_{u,v}\beta$ and recovers with a fixed rate μ . If S_t , I_t , and R_t denote, respectively, the sets of susceptible, infected, and recovered individuals at time t , we assume that $|I_0| = 1$, i.e., a single individual, called

index case, is infectious at time 0. The expected number of infections directly caused by the index case in a completely susceptible population is $R_0^{\text{index}} = \mathbf{E} \left[\sum_{v \neq u} \frac{p_{u,v}\beta}{\mu} \right]$, where \mathbf{E} indicates the average w.r.t. the choice of the index case. As done in [30, 31], we calibrate the model on a common ILI (Influenza Like Illness) by setting the average time to recovery to 3 days and $R_0^{\text{index}} = 1.3$, thus obtaining $\mu = 1/3$ and $\beta \approx 0.0436$.

It may be useful to interpret $p_{u,v}$ as the probability that u and v come into contact and β as the probability that a contact between an infectious agent and a susceptible one results in contagion. Our epidemic model may thus be formulated in terms of a sequence of temporal contact networks $G_I^t = (V, E^t)$, where E^t is the set of contacts occurring at time t of the epidemic simulation, $(u, v) \in E^t$ with probability $p_{u,v}$, and, if $(u, v) \in E^t$, $u \in \mathcal{I}_t$ and $v \in \mathcal{S}_t$, the disease is transmitted from u to v with probability β .

Table 1. The six configurations used for our epidemic model.

Configuration		$p_{u,v}$		
Name	Acronym	$(u, v) \in E_H$	$(u, v) \in E_A$	$(u, v) \notin E$
Homogeneous mixing	HM	$\propto E $	$\propto E $	$\propto E $
Static network	SN	1	1	0
Homogeneous noise	HN	1	0.5	$\propto E - E_H - E_A /2$
Age-based noise	AN	1	0.5	$\propto s_{gu,gv} (E - E_H - E_A /2)$
Distance-based noise	DN	1	0.5	$\propto d(u, v)^{-1} (E - E_H - E_A /2)$
Age- and distance-based noise	ADN	1	0.5	$\propto s_{g(u),g(v)} d(u, v)^{-1} (E - E_H - E_A /2)$

We compare six configurations, summarized in Table 1. In the HM configuration, each G_I^t is an Erdős-Rényi graph, whereas in the SN configuration $G_I^t = G$ for all t . In all other configurations, household relations induce a static frame of *daily* contacts ($p_{u,v} = 1$), acquaintance relations (e.g., between close friends or co-workers) induce *frequent* contacts ($p_{u,v} = 0.5$). In contrast, *fortuitous* contacts (e.g., due to transportation, restaurants or stores) occur between agents not having any stable social relation. The frequency of fortuitous contacts depends on: the age-based social mixing in AN, the geographic distance in DN, both in ADN, neither in HN. By construction, all configurations satisfy $\sum_{u < v} p_{u,v} = |E|$, where $|E|$ is the number of edges of G^1 . It follows that both the expected density of G_I^t and the expected number of potential contagion events $\sum_{u < v} p_{u,v}\beta$ are fixed, thus making the six configurations comparable.

Our epidemic simulations are subject to statistical fluctuations, but the fluctuations induced by different network instances are negligible². We therefore ran 100 identical simulations on a fixed social network instance, so that all fluctuations are due to epidemic dynamics. With a single index case, the fraction of population involved in the outbreak follows a bimodal distribution. The first of

¹ [23] has no similar condition, but our noise is equivalent to theirs in the limit $N \rightarrow \infty$.

² These tests, omitted here due to space limitations, may be made available on request.

the two peaks corresponds to processes that die out in early stages of the epidemic, while the second corresponds to genuine outbreaks that involve a large fraction of the population. The two peaks are well separated at a fraction of around 25% of the population involved so that we can easily discard simulations resulting in early extinction. In the following sections, we only consider the results obtained in the ≈ 30 experiments that actually gave rise to an epidemic outbreak.

3 Epidemic Threshold, Evolution and Reproduction Number

By simulating the spread of an infectious disease, we may understand how susceptible a territory or population is to the occurrence of an epidemic outbreak. This is usually done by looking at: (i) the evolution of the number of infected and recovered individuals at time t , respectively denoted I_t and R_t ; (ii) the epidemic threshold β_c , defined as the critical value for the transmission parameter β after which the fraction of recovered individuals is asymptotically finite; (iii) the epidemic reproduction number $R(t)$, which measures the expected number of new infections caused by a single individual that becomes contagious at time t , during the entire course of his/her infectiousness period; (iv) the basic epidemic reproduction number R_0^{Index} , corresponding to the expected number of infections caused by the index case in an entirely susceptible population. All of these quantities are clearly related to the contact patterns, which, in our case, are described by the degree distribution of the temporal network of contacts G_I^t .

By comparing the degree distribution of G_I^t (Fig. 1a) and the evolution of I_t and R_t (Figs. 1b and 1c), we immediately notice three clusters: (i) under the assumption of homogeneous mixing (HM configuration) the degrees follow a Poisson distribution and the epidemic spread is slower and overall less pervasive than in any data-driven and more realistic configuration; (ii) considering a combination of recurrent contacts, induced by the static social network, and of age-homogeneous fortuitous contacts, whether distance-based or not (HN and DN), produces a skewed degree distribution that causes a sharp acceleration of the outbreak and a slight increase of the total number of cases; (iii) if instead, all contacts are governed by age-based preferences (SN, AN, and ADN), the degree distribution has a longer right tail, and we observe an even earlier peak in the number of infected and an even greater attack rate.

$R(t)$ provides another way to look at the evolution of the epidemic, especially in view of the implementation of potential mitigation strategies. Figure 1d confirms what already emerged in Figs. 1b and 1c: the evolution of the epidemic is almost entirely controlled by the degree distribution of the temporal network of contacts, with the amount and nature of fortuitous contacts playing a major role, but with the dependence of these contacts on distance being almost irrelevant. Figure 1e shows that R_0^{Index} is generally greater in data-driven and realistic configurations, characterized by heterogeneous patterns of contacts. Since we considered only simulations in which an epidemic occurred ($> 25\%$ of

recovered, as discussed in Sect. 2), Fig. 1e highlights how the emergence of an epidemic outbreak is generally associated to the index case being a socially active individual.

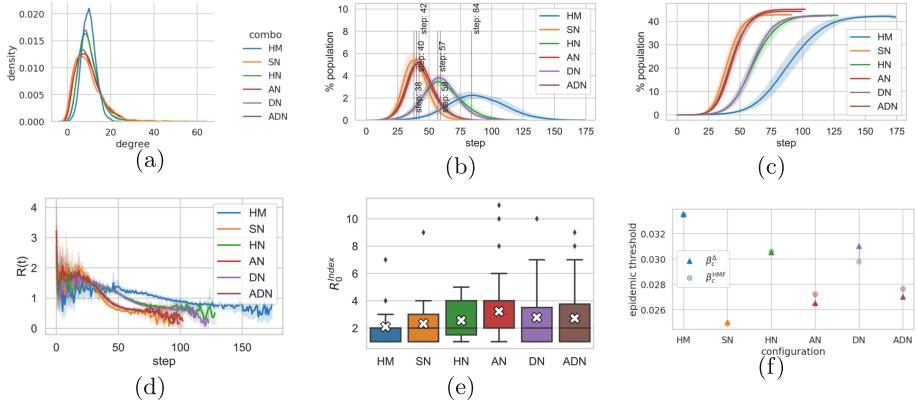


Fig. 1. For all six configurations: (a) degree distribution of the temporal networks G_I^t ; (b,c) evolution of the number of infected (I_t) and recovered (R_t) individuals; (d) evolution of the reproduction number $R(t)$; (e) distribution of the empirical basic reproduction number; (f) empirical epidemic threshold.

The analytical computation of the epidemic threshold for the SIR model on networks is generally based either on the heterogeneous mean-field (HMF) theory, or on the analogy with the bond percolation, leading to expressions for β_c that depend on the first and second moment of the degree distribution. In [23] the authors estimate β_c in a population with two levels of mixing approximating the initial stages of the epidemic with a two-types branching process. To the best of our knowledge, however, no analytical expression is known for our general formulation with multiple levels of mixing – nor for the equivalent formulation in terms of temporal networks. We determined the threshold empirically as in [32], i.e., as $\beta_c^\Delta = \arg \max \Delta(\beta)$, where $\Delta(\beta) = \frac{\sqrt{\langle \rho^2 \rangle - \langle \rho \rangle^2}}{\langle \rho \rangle}$ is the epidemic variability and $\rho = \rho(\beta)$ is the attack rate. Only for the two extreme cases, the HM and SN configurations, the HMF approximation $\beta_c^{\text{HMF}} = \frac{\langle \text{deg} \rangle}{\langle \text{deg}^2 \rangle - \langle \text{deg} \rangle}$ is theoretically valid – where $\langle \text{deg} \rangle$ and $\langle \text{deg}^2 \rangle$ are computed based on the degree distribution of the temporal networks G_I^t . We found, respectively, $\beta_c^{\text{HMF}}(\text{HM}) \approx 0.0335$ and $\beta_c^{\text{HMF}}(\text{SN}) = 0.0249$, in perfect agreement with β_c^Δ . Not surprisingly, the entirely static network SN and the network with HM lie at the two ends of the spectrum, raising doubts on their adequacy to describe realistic contact models. The other four configurations only differ for the way fortuitous contacts are drawn: imposing a dependence on age makes the system more vulnerable to epidemic outbreaks, while assuming a dependence on distance makes it slightly more resistant.

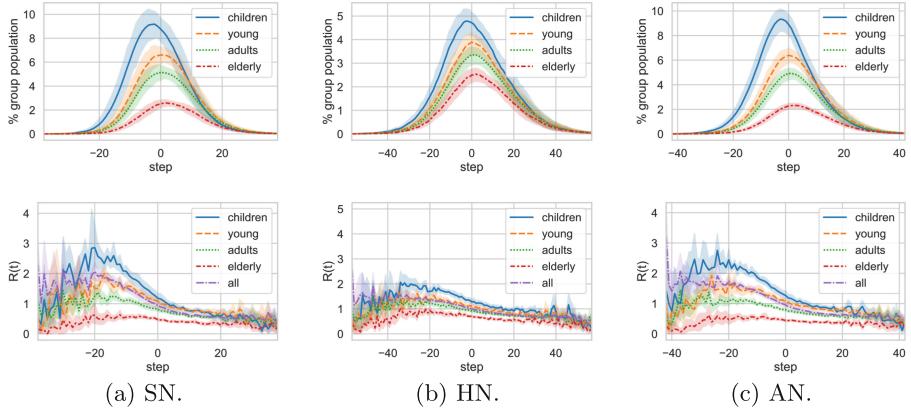


Fig. 2. For the three models SN (a), HN (b) and AN (c): evolution of the number of infected individuals in each age-group and of the epidemic reproduction number for each age-group.

To gain a better understanding of the importance of an age-stratified population, Fig. 2 shows the time evolution of the number of infected individuals I_t and of the reproduction number $R(t)$ for the four age-groups, shifted with respect to the global epidemic peak, for the SN, HN and AN models (HM is trivial, DN and ADN are similar to HN and AN, respectively). Children and young adults seem to drive the epidemic and to be more impacted, as expected given their greater average degree and internal cohesion. The fact that the SN and AN configurations are almost indistinguishable suggests that the diffusion of the disease in different age-groups is dominated by the social mixing, with the recurrence of contacts playing a minor role.

4 Geographic Spreading and Predictability

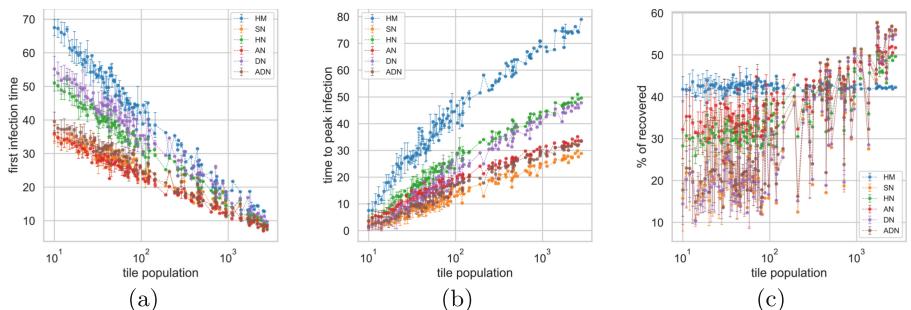


Fig. 3. For each model configuration, as a function of the tile population: (a) mean time of first infection in the tile; (b) mean time interval between first infection and epidemic peak; (c) attack rate. Averages are across equally populated tiles.

A primary goal of data-informed epidemic models is gaining insights into the mechanisms driving the geographic spread of the disease. To assess the role of population density, in Fig. 3 we show the time of the first infection in a tile, the time interval between the contagion entering a tile and the epidemic peak within the tile, and the attack rate (i.e., fraction of recovered at the end of the epidemic), all as a function of the tile population (or, equivalently, the density, our tiles covering a fixed area). The time of the first infection (Fig. 3a) decreases linearly with the logarithm of the population. The infection reaches densely populated tiles in 10 to 15 days, regardless of the configuration, whereas for scarcely populated tiles different ways to model the contact patterns yield very different estimates for the time of the first infection and, hence, for the available time span by which any mitigating actions aimed at preserving low-density areas must be put in place. Both the time to reach the peak of infections (Fig. 3b) and, more notably, the attack rate (Fig. 3c, except for the HM configuration for which the tile of residence is irrelevant) seem to present two regimes: a sub-logarithmic time to the peak and a constant attack rate for tiles with less than ≈ 100 inhabitants; an approximately logarithmically increasing time to the peak and attack rate for densely populated tiles. This behavior – which requires further investigation – may be related to the different ratio of inter- and intra-tile infections in areas with different population density, which may also explain why Fig. 3c is the only case in which the SN, DN and ADN configurations, united by the fact that contacts with individuals living in other tiles are less likely, show an analogous trend: a generally lower attack rate in scarcely populated tiles, compensated by a greater attack rate in high-density tiles.

To understand how the disease travels in the city and whether it follows predictable paths, we focus on the time evolution of three quantities: the fraction of infected tiles, the normalized entropy of the local prevalence vector and the overlap function measuring the similarity between two different outbreak realizations [10]. Formally, let: N_j be the number of inhabitants of tile j ; $I_j(t)$ be the number of infected individuals in tile j at time t ; $\mathbf{q}(t)$ be the normalized vector of prevalence of infected individuals in each tile, with components $q_j(t) = \frac{I_j(t)}{N_j \sum_l I_l(t)/N_l}$; $i(t) = \sum_j \frac{I_j(t)}{N_j}$ be the total epidemic prevalence at time t and $\mathbf{i}(t) = (i(t), 1 - i(t))$; $\boldsymbol{\pi}(t)$ be the vector of probabilities $\pi_j(t) = \frac{I_j(t)}{\sum_l I_l}$ that an infected individual is in tile j at time t ; $\theta^{a,b}(t) = \text{sim}_H(\mathbf{i}^a(t), \mathbf{i}^b(t)) \cdot \text{sim}_H(\boldsymbol{\pi}^a(t), \boldsymbol{\pi}^b(t))$ be the overlap between two different outbreak realizations a and b , where $\text{sim}_H(\mathbf{x}, \mathbf{y}) = \sum_j \sqrt{x_j y_j}$ is the Hellinger affinity. The fraction of infected tiles, defined as $\tau(t) = \frac{|\{j: I_j(t) > 0\}|}{T}$, measures the geographic prevalence of the infection at time t . The normalized entropy $H(t) = -\frac{1}{\log T} \sum_j q_j(t) \log q_j(t)$ of $\mathbf{q}(t)$ is a measure of the level of geographic heterogeneity of the disease prevalence (and, hence, of the riskiness) at time t . The predictability of the system at time t , empirically computed as the distribution of the overlap $\theta^{a,b}(t)$ over all pairs of simulations a, b , measures the similarity between different realizations.

In Fig. 4 we plot $\tau(t)$, $H(t)$ and $\Theta^{a,b}(t)$ (average with 95% confidence interval over a, b) for epidemic simulations in which the index case is chosen uniformly at random in the entire population (Fig. 4a), in a central tile (Fig. 4b) or in a peripheral tile (Fig. 4c). To discount the overall speed of propagation of the infection and make the curves directly comparable, the time scale has been normalized. The fraction of infected tiles is always <0.7 and the AN and DN configurations differ from all others for having, respectively, a visibly greater and lower value of $\tau(t)$ around the epidemic peak. This means that fortuitous contacts tend to favor or contrast the geographic spread of the infection based on whether such contacts are age- or distance-dependent. The normalized entropy never reaches 1 and is significantly <1 for most of the time in all configurations, meaning that the epidemic behavior is geographically heterogeneous and some tiles are, broadly speaking, more dangerous than others. The evolution of the epidemic is only predictable in the proximity of the epidemic peak, which is also when $\tau(t)$ and $H(t)$ are maximal, i.e., when the infection is already highly prevalent in most of the city. At the same time, the fact that the overlap is large close to the peak tells us that we may try to predict which areas are somehow more dangerous at the peak. Even when the tile of origin of the infection is fixed (Figs. 4b and 4c), the predictability drops very quickly at the beginning of the epidemic, to then follow a similar trend to the case where the index case is chosen uniformly at random (Fig. 4a). The behavior of $H(t)$ and $\Theta^{a,b}(t)$ does not differ appreciably among configurations, except that the system is generally more homogeneous and predictable with the increase of the constraints.

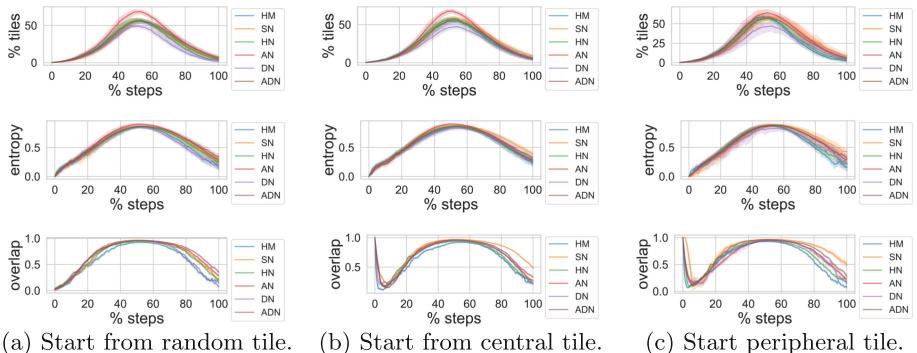


Fig. 4. For all six configurations, fraction of infected tiles (top), normalized entropy of the local prevalence (center) and predictability (bottom) and when the index case is chosen: (a) at random in the entire population; (b) at random in a central tile; (c) at random in a peripheral tile. The time scale is normalized to make all curves comparable.

5 Conclusion

To contribute to the understanding of epidemic outbreaks in an urban territory, we defined an individual-based SIR model for a data-driven synthetic population of agents connected by a social network of stable social relations. Our model assumes that an individual has daily contacts with her household members, frequent contacts with a small network of acquaintances (e.g., friends or coworkers), and fortuitous contacts with the rest of the population. We considered four configurations that differ for the dependence of fortuitous contacts on demographic and/or geographic features of the host population, along with two benchmark configurations respectively characterized by the assumption of a random network (homogeneous mixing), or a static contact network.

We found that age-structure in contact patterns causes a right-skewed degree distribution that makes the epidemic faster and more pervasive. Imposing that the interaction frequency decays with distance as d^{-1} , as suggested by previous empirical studies, seems to have little effect on the high-level diffusion patterns. A faster decay may result in a more significant impact [33], but it would require empirical justification. The distance still plays a role in determining the frequency of inter-tile infections, which affects the relation between the population of a tile and its attack rate. Population density has a considerable impact on the local incidence of the epidemic and its temporal evolution. In particular, the attack rate shows two distinct behaviors for low/high density populated tiles. Our result echoes the empirical finding of two different diffusion regimes on the country/city scale with low/high population density [20,34]. This analogy could indicate a pattern of hierarchical spatial diffusion even in urban areas when population heterogeneity is taken into account, but further investigation is needed.

We noticed the absence of backbones capable of defining preferred epidemic pathways. This fact is not entirely surprising since all our configurations present heterogeneous connectivity patterns and a wide range of alternative and equivalent travel routes for the infection. In data-driven meta-population models, such as the one used in [10] to describe the role of the airline transportation network in global epidemics, the local and global dynamics of contagion are well separated, and, at the broader scale, the high number of possible spreading channels is compensated by the presence of dominant connections in traffic flow. This is not the case for our model of urban contacts, where the only stratification of contacts is induced by the discretization of the distance function $d(u, v)$ used to create the acquaintance network, and, possibly, to establish the likelihood of fortuitous contacts. Understanding whether the absence of dominant epidemic pathways is inherently due to the different scale of the analysis or it can be addressed by informing the model with additional data (e.g., mobility, transportation, phone calls) is a primary direction for future work.

References

1. Kermack, W.O., McKendrick, A.G.: Contributions to the mathematical theory of epidemics-I. Bull. Math. Biol. **53**(1–2), 33–55 (1991)

2. Hethcote, H.W.: The mathematics of infectious diseases. *SIAM Rev.* **42**(4), 599–653 (2000)
3. Mossong, J., et al.: Social contacts and mixing patterns relevant to the spread of infectious diseases. *PLOS Med.* **5**(3), 1–1 (2008)
4. Cattuto, C., Van den Broeck, W., Barrat, A., Colizza, V., Pinton, J.-F., Vespignani, A.: Dynamics of person-to-person interactions from distributed RFID sensor networks. *PloS one* **5**(7), e11596 (2010)
5. Mistry, D., et al.: Inferring high-resolution human mixing patterns for disease modeling (2020). arXiv preprint [arXiv:2003.01214](https://arxiv.org/abs/2003.01214)
6. Eubank, S., et al.: Modelling disease outbreaks in realistic urban social networks. *Nature* **429**(6988), 180–184 (2004)
7. Merler, S., Ajelli, M.: The role of population heterogeneity and human mobility in the spread of pandemic influenza. *Proc. Royal Soc. B Biol. Sci.* **277**(1681), 557–565 (2009)
8. Chang, S., et al.: Mobility network models of covid-19 explain inequities and inform reopening. *Nature* **589**(7840), 82–87 (2021)
9. Hufnagel, L., Brockmann, D., Geisel, T.: Forecast and control of epidemics in a globalized world. *Proc. Natl. Acad. Sci.* **101**(42), 15124–15129 (2004)
10. Colizza, V., Barrat, A., Barthélémy, M., Vespignani, A.: The role of the airline transportation network in the prediction and predictability of global epidemics. *Proc. Natl. Acad. Sci.* **103**(7), 2015–2020 (2006)
11. Tizzoni, M., et al.: On the use of human mobility proxies for modeling epidemics. *PLoS Comput. Biol.* **10**(7), e1003716 (2014)
12. Colizza, V., Barrat, A., Barthélémy, M., Vespignani, A.: Predictability and epidemic pathways in global outbreaks of infectious diseases: the SARS case study. *BMC Med.* **5**(1), 1–13 (2007)
13. Cazelles, B., Champagne, C., Dureau, J.: Accounting for non-stationarity in epidemiology by embedding time-varying parameters in stochastic models. *PLoS Comput. Biol.* **14**(8), e1006211 (2018)
14. Gupta, S., Anderson, R.M., May, R.M.: Networks of sexual contacts: implications for the pattern of spread of HIV. *AIDS (London, England)* **3**(12), 807–817 (1989)
15. Newman, M.E.J.: Spread of epidemic disease on networks. *Phys. Rev. E* **66**(1), 016128 (2002)
16. Keeling, M.J., Eames, K.T.D.: Networks and epidemic models. *J. Royal Soc. Interf.* **2**(4), 295–307 (2005)
17. Chakrabarti, D., Wang, Y., Wang, C., Leskovec, J., Faloutsos, C.: Epidemic thresholds in real networks. *ACM Trans. Inf. Syst. Secur. (TISSEC)* **10**(4), 1–26 (2008)
18. Pastor-Satorras, R., Castellano, C., Van Mieghem, P., Vespignani, A.: Epidemic processes in complex networks. *Rev. Mod. Phys.* **87**(3), 925 (2015)
19. Charu, V., et al.: Human mobility and the spatial transmission of influenza in the united states. *PLoS Comput. Biol.* **13**(2), e1005382 (2017)
20. Grenfell, B.T., Bjørnstad, O.N., Kappey, J.: Travelling waves and spatial hierarchies in measles epidemics. *Nature* **414**(6865), 716–723 (2001)
21. Li, R., Richmond, P., Roehner, B.M.: Effect of population density on epidemics. *Physica A Stat. Mech. Appl.* **510**, 713–724 (2018)
22. Ball, F., Neal, P.: A general model for stochastic sir epidemics with two levels of mixing. *Math. Biosci.* **180**(1–2), 73–102 (2002)
23. Ball, F., Neal, P.: Network epidemic models with two levels of mixing. *Math. Biosci.* **212**(1), 69–87 (2008)
24. Kiss, I.Z., Green, D.M., Kao, R.R.: The effect of contact heterogeneity and multiple routes of transmission on final epidemic size. *Math. Biosci.* **203**(1), 124–136 (2006)

25. Bondarenko, M., Kerr, D., Sorichetta, A., Tatem, A.: Census/projection-disaggregated gridded population datasets for 189 countries in 2020 using built-settlement growth model (bsgm) outputs (2020)
26. Guarino, S., et al.: Inferring urban social networks from publicly available data. *Fut. Internet* **13**(5), 108 (2021)
27. Willem, L., Van Hoang, T., Funk, S., Coletti, P., Beutels, P., Hens, N.: SOCRATES: an online tool leveraging a social contact data sharing initiative to assess mitigation strategies for COVID-19. *BMC Res. Notes* **13**(1), 1–8 (2020)
28. Caldarelli, G., Capocci, A., De Los Rios, P., Muñoz, M.A.: Scale-free networks from varying vertex intrinsic fitness. *Phys. Rev. Lett.* **89**, 258708 (2002)
29. Guarino, S., et al.: A model for urban social networks. In: Paszynski, M., Kranzmüller, D., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M.A. (eds.) ICCS 2021. LNCS, vol. 12744, pp. 281–294. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77967-2_23
30. Biggerstaff, M., Cauchemez, S., Reed, C., Gambhir, M., Finelli, L.: Estimates of the reproduction number for seasonal, pandemic, and zoonotic influenza: a systematic review of the literature. *BMC Infect. Dis.* **14**(1), 1–20 (2014)
31. Liu, Q.-H., Ajelli, M., Aleta, A., Merler, S., Moreno, Y., Vespignani, A.: Measurability of the epidemic reproduction number in data-driven contact networks. *Proc. Natl. Acad. Sci.* **115**(50), 12680–12685 (2018)
32. Shu, P., Wang, W., Tang, M., Do, Y.: Numerical identification of epidemic thresholds for susceptible-infected-recovered model on finite-size networks. *Chaos Interdisc. J. Nonlinear Sci.* **25**(6), 063104 (2015)
33. Medo, M.: Contact network models matching the dynamics of the covid-19 spreading. *J. Phys. A Math. Theor.* **54**(3), 035601 (2020)
34. Viboud, C., Bjørnstad, O.N., Smith, D.L., Simonsen, L., Miller, M.A., Grenfell, B.T.: Synchrony, waves, and spatial hierarchies in the spread of influenza. *Science* **312**(5772), 447–451 (2006)

Dynamics on/of Networks



From Mean-Field to Complex Topologies: Network Effects on the Algorithmic Bias Model

Valentina Pansanella^{1,3}(✉), Giulio Rossetti³, and Letizia Milli^{2,3}

¹ Scuola Normale Superiore, Pisa, Italy

valentina.pansanella@sns.it

² Department of Computer Science, University of Pisa, Pisa, Italy

milli@di.unipi.it

³ KDD Lab ISTI-CNR, Pisa, Italy

{valentina.pansanella,giulio.rossetti,letizia.milli}@isti.cnr.it

Abstract. Nowadays, we live in a society where people often form their opinion by accessing and discussing contents shared on social networking websites. While these platforms have fostered information access and diffusion, they represent optimal environments for the proliferation of polluted contents, which is argued to be one of the co-causes of polarization/radicalization. Moreover, recommendation algorithms - intended to enhance platform usage - are likely to augment such phenomena, generating the so called *Algorithmic Bias*. In this work, we study the impact that different network topologies have on the formation and evolution of opinion in the context of a recent opinion dynamic model which includes bounded confidence and algorithmic bias. Mean-field, scale-free and random topologies, as well as networks generated by the Lancichinetti-Fortunato-Radicchi benchmark, are compared in terms of opinion fragmentation/polarization and time to convergence.

Keywords: Opinion dynamics · Complex networks · Algorithmic bias

1 Introduction

One of the most analyzed phenomena on online social networks is the tendency to observe political polarization, e.g., online discourse's tendency to divide users into opposite political factions not aiming at reaching any form of synthesis. Polarization not only emerges in political debates, but it also often characterizes a variety of controversial topics, and, in some cases, it may affect policymaking and society.

The tendency to observe political polarization, e.g., online discourse's tendency to divide users into opposite political factions has captured great interest recently [2], affecting also polarization on a wide variety of controversial topics and eventually policymaking and society.

Among the various causes of this phenomenon, there is considered to be the rise of social media, in particular because of the presence of personalization/recommendation algorithms [13]. Even if their initial intent was to maximize platform usage and users' engagement, these algorithms end up working as a reinforcement bias for online users' opinions, neglecting them access/confront with narratives different from their own.

The field of opinion dynamics aims at understanding the opinion formation, evolution, and eventually stabilization in groups of interacting agents. Typically, such models consider a finite number of interacting people (*agents*); each agent has its own opinion, discrete or continuous, which can vary over time, according to rules to explain the change due to interactions with other agents.

The idea of studying human behavior in the same way that physics is studied dates back to the 18th century with the work of the philosopher David Hume [8]. This science is referred to as "sociophysic" [11], and it draws on the idea that general laws are describing human societies and human behavior; therefore, the statistical analysis can be performed on human qualities to explain human behavior. The advent of Big Data stimulated a renewed interest in this field, first regarding financial markets and then exploiting data from the internet and online social networks.

The last decade witnessed the introduction of a wide variety of models that start from the milestones [3, 4, 6, 7] and extend them, considering different socio-logical theories that try to incorporate the new characteristics of today's society that influence group dynamics [12]. With the development of online social networks, more and more works tried to include their specific characteristics in the opinion dynamics model to get closer to reality. Filtering algorithms [13], fake news, underlying complex networks with specific topologies and communities, are considered essential elements to implement a good opinion dynamics model, and researchers are surpassing the idea of complete network structures and synchronous interactions.

In this paper, we investigate the expected effects of algorithmic bias in a networked population. Moving from the results discussed in [13] where a mean-field context is assumed (e.g., all individuals can interact among them without any social restrictions), we aim to study the effect that different network topologies have on opinion formation and evolution when in the presence of a filtering algorithm.

To such extent, and to allow results reproducibility, we focus our analysis on well-known network models, namely Erdős-Rényi [5], Barabási-Albert [1] and Lancichinetti-Fortunato-Radicchi benchmark [9] graphs (henceforth referred to as LFR graphs).

Adopting such controlled environments, used to simulate the social structure among a population of interacting individuals, we analyze the behaviors of the Algorithmic Bias model [13] (e.g., an extension of the Deffuant-Weisbuch [3] opinion dynamics model) and discuss the role of graph properties on the observed simulations results.

The paper is organized as follows. In Sect. 2 we introduce the algorithmic bias model, and we describe our experimental workflow. Section 3 discusses the main finding of our simulations, finally Sect. 4 concludes the paper while opening to future investigations.

2 Algorithmic Bias: From Mean-Field to Complex Topologies

Considering such a complex scenario, the present work aims to deepen the AB model's behavior analysis and test it on different network topologies.

Nowadays, online social networks have become the primary source of information and an excellent platform for opinion exchanges. However, the flow of content that each user sees is organized by algorithms that are built to maximize platform usage: from this, it comes to the hypothesis that there is an algorithmic bias (also called algorithmic segregation) since these contents are selected based on users' precedent actions on the platform or the web, reinforcing the human tendency to interact with content confirming their beliefs (confirmation bias).

To introduce in the study of opinion dynamics the idea of a recommender system generating an algorithmic bias, we started from a recent extension of the well-known Deffuant-Weisbuch model (DW-model henceforth), proposed in [13] (Algorithmic Bias model or AB model, henceforth).

In the Algorithmic Bias model, we have a population of N agents, where each agent i has a continuous opinions $x_i \in [0, 1]$. At every discrete time step the model randomly select a pairwise (i, j) , and, if their opinion distance is lower than a threshold ϵ , $|x_i - x_j| \leq \epsilon$, then the two agents change their opinion according to the following rule:

$$\begin{aligned} x_i(t+1) &= x_i + \mu(x_j - x_i) \\ x_j(t+1) &= x_j + \mu(x_i - x_j). \end{aligned} \quad (1)$$

The parameter $\epsilon \in [0, 1]$ models the population's confidence bound, and it is assumed to be constant among all the agents. The parameter $\mu \in (0, 0.5]$ is a convergence parameter, modeling the strength of the influence the two individuals have on each other or, in other words, how much they change their opinion after the interaction. The numerical simulations of this model show that the qualitative dynamic is mainly dependent on ϵ : as ϵ grows, the number of final opinion clusters decreases. As for μ and N , these parameters tend to influence only the time to convergence and the final opinion distribution width.

The AB model introduces another parameter to model the algorithmic bias: $\gamma \geq 0$. This parameter represents the filtering power of a generic recommendation algorithm: if it is close to 0, the agent has the same probability of interacting with all its peers. As γ grows, so does the probability of interacting with agents holding similar opinions, while the probability of interacting with those who hold distant opinions decreases. Therefore, this extended model modifies the rule to choose the interacting pair (i, j) to simulate a filtering algorithm's presence. An

agent i is randomly picked from the population, while j is chosen from i 's peers according to the following rule:

$$p_i(j) = \frac{d_{ij}^{-\gamma}}{\sum_{k \neq i} d_{ik}^{-\gamma}} \quad (2)$$

where $d_{ij} = |x_i - x_j|$ is the opinion distance between agents i and j , so that for $\gamma = 0$ the model goes back to the DW-model. When two agents interact, their opinions change if and only if the distance between their opinions is less than the parameter ϵ , i.e. $|x_i - x_j| \leq \epsilon$, according to Eq. 1. In [13] the discussed AB model results focus only on the mean-field scenario, i.e., the authors assume a complete graph as the underlying social structure.

On online social networks, each agent will likely interact only with whom they follow or with their friends.

Our goal is to verify if/how networks' structure exacerbates the polarization and fragmentation generated by the selection bias's presence. We want to verify if moving from a complete network with L_{max} links to a network with $L \ll L_{max}$ links and a predetermined topology influences the final simulation state, making it harder for the population to reach a consensus or ultimately preventing it.

To such an extent, we first replicate the work of [13] on a complete network (to define a reliable baseline), and after that, we show the AB evolution on two different graph models: random [5] and a scale-free [1] network. We extend the experimentation carried out in [13] because the mean-field assumption neglect key properties of real networks, such as the sparsity and the small-world phenomenon (captured by the ER model [5]) and the scale-free degree distribution (captured by the BA model [1]).

Moreover, to understand the impact of realistic mesoscale network topology (e.g., presence of communities) on opinion evolution, we also test the AB model against a network generated through the LFR benchmark [9]. Our research aims to investigate whether, in realistic environments, opinions remain trapped inside communities or not and which are the effects of different topologies on the steady-state of the modeled dynamic process, e.g., whether they facilitate/counteract polarization/fragmentation or promote consensus.

In all scenarios, we set the number of nodes $N = 250$. For the ER network, we fix the p parameter (probability to form a link) to 0.1 (thus imposing a *supercritical* regime, as expected from a real-world network); we obtain a random network composed of a single giant component with an average degree of 24.94. In the BA network, we set the k parameter (number of edges to attach from a new node to existing nodes) to 5, thus creating a network with an average degree equal to 9.8.

We generated nine different networks using the LFR benchmark ($N = 250$). The parameters used for its construction have been set as follows:

- power-law exponent for the degree distribution, $\gamma = 3$;
- power-law exponent for the community size distribution, $\beta = 1.5$;
- fraction of intra-community edges incident to each node,

$$\mu_{LFR} \in \{0.1, 0.5, 0.9\};$$

- average degree of nodes, $\langle k \rangle = 10$;
- minimum community size $min_s = 50$, thus losing the power-law community size distribution and generating 4 communities of similar sizes in the end.

The parameter μ_{LFR} controls the number of edges between communities, thus reflecting the network's amount of noise. Therefore, the network with $\mu_{LFR} = 0.1$ has better-defined communities than the one generated with $\mu_{LFR} = 0.9$.

3 Experimental Analysis and Results

Like in [13], to avoid undefined operations in Eq. 2, when $d_{ik} = 0$ we use a lower bound $d_\epsilon = 10^{-4}$. The simulations are designed to stop when the population reaches an equilibrium, i.e., the cluster configuration will not change anymore, even if the agents keep exchanging opinions. We also set an overall maximum number of iterations at 10^5 . To account for the model's stochastic nature, we compute the average results over 10 independent executions for each configuration, where the initial opinion distribution is always drawn from a random uniform probability distribution in $[0,1]$. To better understand the differences in the final state concerning the different topologies considered, we study the model on all networks for different combinations of the parameters. We are interested in whether, parameters being equal, the different topology influences the final cluster configuration enhancing polarization and fragmentation, but also the dynamics of the process, by slowing down the convergence or reducing the density of the final opinion clusters. In the simulations, we tested the model on every possible combination of the parameters over the following values:

- ϵ takes a value from 0.2 to 1.0 with the step of 0.1.
- γ takes value from 0 to 2.0 with the step of 0.2; for $\gamma = 0$ the model becomes the DW-model.
- $\mu = 0.5$, so whenever two agents interact, if their opinions are close enough, they update to the pair's average opinion.

The AB model implementation used to carry out our experiments is the one provided by the NDlib [10] Python library¹.

For the simulations of the AB model on the LFR benchmark networks, instead, we tested the model over the following values:

- $\epsilon \in \{0.2, 0.3\}$. We impose this choice because in the mean-field, for these values, the number of clusters grows with increasing gamma, and we obtain a situation of polarization and fragmentation;
- $\gamma \in \{0.0, 0.5, 1.0, 1.5, 2.0\}$;
- $\mu = 0.5$. With this value, whenever two agents interact, if their opinions are close enough, they update to the pair's average opinion.

¹ NDlib: <http://ndlib.rtfd.io>.

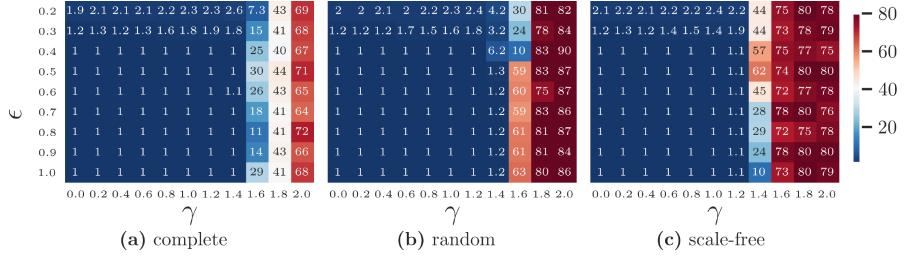


Fig. 1. Average number of clusters across topologies. The figure displays the average number of clusters as a function of ϵ and γ , over 10 runs. We show the results both for γ from 0 to 2.0 with step of 0.2 and ϵ from 0.2 to 1.0 with step 0.1 respectively for a complete network (a) a random network (b) and a scale-free network (c)

Average Number of Clusters. To analyze the results of the simulation we start by taking into account the number of final opinion clusters in the population, to understand the degree of fragmentation that the different combination of the parameters produce. This value indicates how many peaks there are in the final distribution of opinions and provides a first approximation of whether a consensus can be obtained or not. To compute the effective number of clusters, accounting for the presence of major and minor ones, we use the cluster participation ratio, as in [13]:

$$C = \frac{(\sum_i c_i)^2}{\sum_i c_i^2} \quad (3)$$

where c_i is the dimension of the i th cluster, i.e., the fraction of population we can find in that cluster. In general, for m clusters, the maximum value of the participation ratio is m and is achieved when all clusters have the same size, while the minimum can be close to 1, if one cluster contains most of the population and a very small fraction is distributed among the other $m - 1$.

From Fig. 1 we can see that the behavior of the model across the different network topologies is very similar: the growth of the confidence bound ϵ allows the population that initially ended up as polarized to reach a full and perfect consensus, at least up to a certain value of the algorithmic bias γ . The experiments on the three different networks show how the population either converges to one or a few significant clusters or fragments over a wide range of opinions, when γ is above a certain threshold: the final state shows tens of clusters populated by few agents that cannot merge in the time span allowed in these experiments.

Even in the mean-field for $\gamma \geq 1.6$ the effect of the algorithmic bias is too strong to be mitigated by an increment in the bounded confidence parameter ϵ . The total number of clusters grows with γ from values around 10 to values around 70.

However, the population only has 10^5 iterations to reach convergence and in some cases the process reaches this bound without having reached equilibrium, like we will see later in this paragraph. In [13], instead, the maximum number of

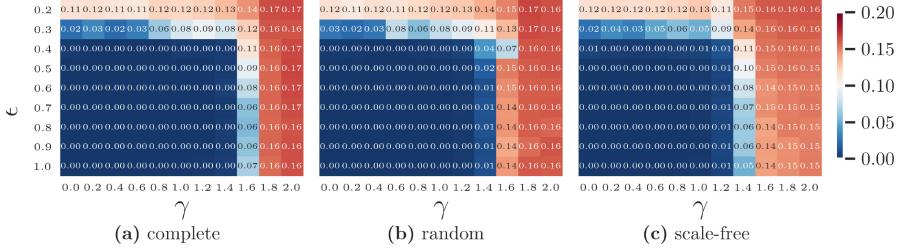


Fig. 2. Average opinion distance across topologies. The figure displays the average pairwise opinion distance as a function of ϵ and γ , over 10 runs. We show the results both for γ from 0 to 2.0 with step of 0.2 and ϵ from 0.2 to 1.0 with step 0.1 respectively for a complete network (a) a random network (b) and a scale-free network (c)

iterations was set to 10^7 , allowing the population to always reach a steady-state. While analysing what happens when time goes to infinite is important, it's also important to understand how the final status may change with a much shorter dynamic. The present results could mean that consensus - even if theoretically possible - may never be reached in a real setting where there is a finite amount of time to discuss a topic and the population may instead remain fragmented.

Considering only $\gamma \leq 1.4$ we can see that up to this value, the results remain the same described in [13]: for $\epsilon \geq 0.4$ a consensus is always reached, even if it tends to become less and less perfect, while for $\epsilon \leq 0.4$ the number of clusters increases with the bias, which brings the population to a polarization of opinions even in situations where the DW-model [3] would have produced a full consensus.

Introducing a different network topology, such as a random network or a scale-free network, however, produces a change in the behaviour for very strong biases. Such a result suggests that a sparser topological structure has a small impact on the observed results until the introduced bias is not strong. However, as the algorithmic filtering grows stronger, the sparsity has a very severe impact, preventing consensus - even in cases where it was observed as a possible outcome in mean-field. Moreover while for γ values below the fragmentation threshold the effective number of clusters is very similar across the three different network topologies, in the fragmented state we can see that in the scale-free case the number of clusters is higher - on average - than in the random case and both show overall higher values with respect to the complete networks. It is not clear how this different behaviour depends on the topology and how it depends on a different average degree and thus total number of links in the networks, but we can assume that the more the sparsity, the more it gets difficult for opinion clusters to merge when the bias limits very much the number of agents to interact with (Fig. 2).

Average Pairwise Distance. To study the degree of polarization/fragmentation, we computed the average pairwise distance between the agents' opinions. Given an agent i with opinion x_i and an agent j with opinion x_j at the end of the diffusion process, the pairwise distance between the two agents is $d_{ij} = |x_i - x_j|$. The average pairwise distance in the final state can be computed as $\frac{\sum_{i=0}^N \sum_{j=0}^N d_{ij}}{N^2}$. In every network, the average opinion distance goes from a minimum value of 0.0, when the population reaches a full consensus and every agent holds the same opinion, to a maximum value of 0.15 – 0.17 when there are tens of opinion clusters in the population. We can observe that such a distance follows the same pattern identified in the number of clusters: it decreases as ϵ grows and grows with γ . However, an important thing to point out is that while the difference in the number of clusters can be very high, the opinion distance differences are not so high between a state with 3 final clusters and a state with 80 final clusters. Indeed, when the opinion distribution is very fragmented, the different clusters tend to get closer to each other. This holds for the three different networks considered in this work.

There are also some cases in the complete network where the average pairwise distance decreases despite the number of clusters in the final state is higher. This result suggests that the final peaks in the opinion distribution are indeed all very close to each other and with a longer simulation a lower level of fragmentation could be reached.

That considered, we can state that the average pairwise distance is suitable to highlight the transition from consensus to polarisation or a limited number of clusters. However, it is not suitable to characterize a growing fragmentation or an intermediate state where there are still many cluster growing closer to each other before merging.

Time to Convergence. Finally, we consider the time to convergence. The time to convergence is measured as the number of iterations (each constituted by N pairwise interactions).

Figure 3 compares the evolution of the time to convergence as a function of ϵ and γ . The three plots all show a similar behavior: the main impact on time to convergence (since μ and N are fixed) is given by γ . In particular, for every value of the parameter ϵ in every network the convergence slows down until it reaches its peak for a certain value of the bias, then the time to convergence starts to decrease as the bias grows.

Mesoscale Structure. We saw from the previous analysis that changing the topology of the network, even with a low average degree and a scale-free structure - doesn't affect much the dynamics. To understand how - instead - the addition of a mesoscale structure may affect the process of opinion diffusion we simulated three different scenarios over the previously described LFR networks and we analyzed the same measures as a function of γ , ϵ and also μ_{LFR} . We fixed three different settings:

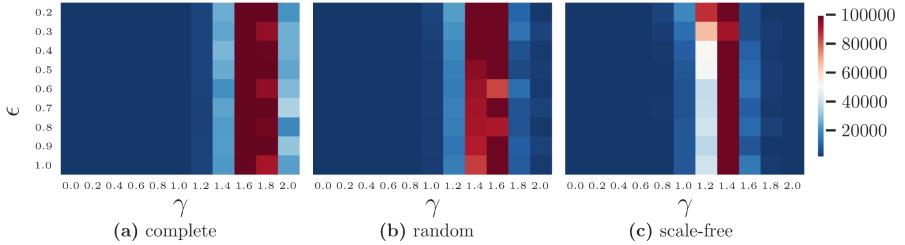


Fig. 3. Average number of iterations to convergence across topologies. The figure displays the average number of iterations to convergence as a function of ϵ and γ , over 10 runs. We show the results both for γ from 0 to 2.0 with step of 0.2 and ϵ from 0.2 to 1.0 with step 0.1 respectively for a complete network (a) a random network (b) and a scale-free network (c)

- i) the opinions are uniformly distributed across the whole population, like on complete, random, and scale-free cases previously analyzed;
- ii) a random mean opinion to each community is assigned, and then the opinions are normally distributed within the community with standard deviation equal 0.01;
- iii) the opinions are normally distributed with predefined means $\in \{0.25, 0.5, 0.75, 1.0\}$ and a standard deviation equal 0.01.

From Fig. 4 it seems that the dynamics remains qualitatively the same as in the previous cases. A higher ϵ foster consensus while as γ grows so does fragmentation. However, even with opinion randomly distributed across population, it seems that the mesoscale structure reduces the fragmenting effects of the bias (Fig. 4(a)–(b)) resulting in a lower number of clusters for very high values of bias. If we start assigning to each community a random mean opinion and distributing opinions across community members with a small variance (Fig. 4(c)–(d)) we can see that fragmentation is overall reduced. When the mean opinions of the communities are more distant than the confidence bound (Fig. 4(e)–(f)) we always obtain four to five final clusters, since different communities cannot merge and eventually, when the selection bias is very strong, some of them split into more than one cluster. However, in the case of polarized communities, if the mean initial opinions are less distant than the confidence bound, the dynamic remains the same: we see a slow rise from polarization to fragmentation as the bias grows.

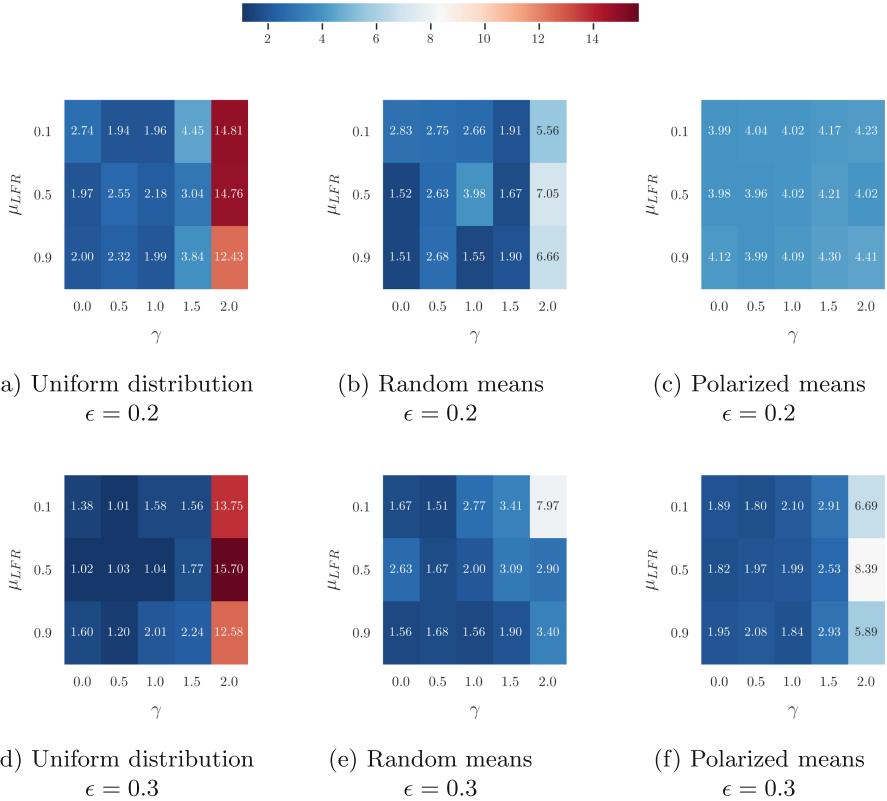


Fig. 4. Average number of clusters for a given value of ϵ as a function of μ_{LFR} and γ .

4 Conclusion

In this work, the Algorithmic Bias model - developed within the framework of bounded confidence - was simulated on complete, random scale-free network topologies. Such an analysis was carried out to discover and characterize the differences that affect model simulation outcomes while moving from a mean-field scenario (as proposed by the original authors) toward a more complex ones.

Algorithmic bias is argued to be an existing factor affecting several (online) social environments. Since interactions occurring among agents embedded in such realities are far from being easily approximated by a mean-field scenario, in our study, we aimed to understand the role played by alternative network topologies on the outcome of biased opinion dynamic simulations.

From our study emerges that the qualitative dynamic of opinions remains substantially in line with what was observed assuming a mean-field context: an increase in the confidence bound ϵ favors consensus. In contrast, the introduction of the algorithmic bias γ hinders it and favors fragmentation. Conversely, both

simulations' time to convergence and opinions fragmentation appears to increase as the topology becomes sparser and the hub emerges. Therefore, our analysis underlines that, alongside the algorithmic bias, the network's density heavily affects the degree of consensus reachable, assuming a population of agents with the same initial opinion distribution.

We also investigated how an underlying community structure affects the dynamics. What emerged is that the community structure enhances the consensus, and a larger algorithmic bias has the only effect of slowing down the convergence process. As already stated by the authors in [13], the initial condition is crucial to determine the final state. Our work showed that polarized communities that are further than the confidence bound cannot converge and that an increasing bias may favor splits into two or more clusters within the same community, even when the starting opinions were very close. As future work, we plan to extend the AB model to cope with more realistic scenarios involving dynamic network topologies and the existence of exogenous factors affecting agents' behaviors.

Acknowledgments. This work is supported by the scheme ‘INFRAIA-01-2018-2019: Research and Innovation action’, Grant Agreement n. 871042 ‘SoBigData++: European Integrated Infrastructure for Social Mining and Big Data Analytics’

References

1. Barabási, A.-L., Albert, R.: Emergence of scaling in random networks. *Science* **286**, 509–512 (1999)
2. Campbell, J.E.: *Polarized: Making Sense of a Divided America*. Princeton University Press, Princeton (2018)
3. Deffuant, G., Weisbuch, G.: Mixing beliefs among interacting agents. *Adv. Complex Syst.* **3**, 87–98 (2000)
4. DeGroot, M.: Reaching a consensus. *J. Am. Stat. Assoc.* **69**, 118–121 (1974)
5. Erdős, P., Rényi, A.: On random graphs. i. *Publicationes Mathematicae* **6**, 290–297 (1959)
6. French, J.R.: A formal theory of social power. *Psychol. Rev.* **63**, 181–194 (1956)
7. Friedkin, N.E.: A formal theory of social power. *J. Math. Sociol.* **12**, 103–126 (1986)
8. Hume, D.: *A Treatise of Human Nature: being an Attempt to introduce the Experimental Method of Reasoning into Moral Subjects. Of the Understanding [volume I]*. John Noon (1739)
9. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78**(4), 146110 (2008)
10. Rossetti, G., Milli, L., Rinzivillo, S., Sirbu, A., Pedreschi, D., Giannotti, F.: Ndlib: a python library to model and analyze diffusion processes over complex networks. *Int. J. Data Sci. Anal.* **5**(1), 61–79 (2018)
11. Schweitzer, F.: Sociophysics. *Phys. Today* **71**, 40 (2018)

12. Sîrbu, A., Loreto, V., Servedio, V.D.P., Tria, F.: Opinion dynamics: models, extensions and external effects. In: Loreto, V., et al. (eds.) *Participatory Sensing, Opinions and Collective Awareness*. UCS, pp. 363–401. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-25658-0_17
13. Sîrbu, A., Pedreschi, D., Giannotti, F., Kertész, J.: Algorithmic bias amplifies opinion fragmentation and polarization: a bounded confidence model. *PloS one* **14**(3), e0213246 (2019)



Towards Control of Opinion Diversity by Introducing Zealots into a Polarised Social Group

Antoine Vendeville^{1(✉)}, Benjamin Guedj^{1,2}, and Shi Zhou¹

¹ Department of Computer Science, Centre for Doctoral Training in Cybersecurity and Centre for Artificial Intelligence, University College London, 90 High Holborn, London WC1V 6LJ, UK

a.vendeville@ucl.ac.uk

² Inria, Lille – Nord Europe Research Centre, Lille, France
<http://antoinevendeville.github.io>

Abstract. We explore a method to influence or even control the diversity of opinions within a polarised social group. We leverage the voter model in which users hold binary opinions and repeatedly update their beliefs based on others they connect with. Stubborn agents who never change their minds (“zealots”) are also disseminated through the network, which is modelled by a connected graph. Building on earlier results, we provide a closed-form expression for the average opinion of the group at equilibrium. This leads us to a strategy to inject zealots into a polarised network in order to shift the average opinion towards any target value. We account for the possible presence of a *backfire effect*, which may lead the group to react negatively and reinforce its level of polarisation in response. Our results are supported by numerical experiments on synthetic data.

Keywords: Voter model · Opinion dynamics · Social networks

1 Introduction

We are interested in controlling opinions on connected networks with arbitrary degree distribution. In recent years, recommendation algorithms on social platforms have greatly enhanced confirmation bias by showing users content that is the most susceptible to match their interests—the so-called “filter bubble” effect [28]. As a consequence more and more isolated, tightly clustered online communities of similar-minded individuals have arisen in various domains such as politics [10, 12, 17], healthcare [2, 21] or science [32]. Because of the so-called *backfire effect*, presenting these users with opposing information might have the adverse effect of reinforcing their prior beliefs [4, 30]. In this paper we provide a simple method to shift diversity of opinions within towards a chosen target level, with and without the presence of a backfire effect.

To this end we rely on the well-known voter model, in which each user holds one of two possible opinions (e.g. liberal or conservative, pro or anti-abortion) and updates it randomly under the distribution of others' beliefs. Independently introduced by Clifford and Sudbury [9] and Holley [20] in the context of particles interaction, this model has since been used to describe in a simple and intuitive manner social dynamics where people are divided between two parties and form their opinion by observing that of others around them. We assume some of the users are stubborn and never change opinion. We call them *zealots* as in [24, 25]. They can represent lobbyists, politicians or activists for example. Long time dynamics and limiting behaviour of such processes have been subject to several studies [25, 27, 34].

To achieve our goal we extend a previous result from the literature. Namely, authors of [25] found an expression for the average number of opinion-1 users at equilibrium in the $n \rightarrow \infty$ limit for a fully-connected network. Here we prove that their result also holds on expectation for any connected graph, assuming the placement of zealots is done uniformly at random. This allows us to find the optimal number of zealots to inject in a polarised community in order to reach any target average opinion at equilibrium, with and without the presence of a backfire effect. This effect we model simply by assuming that the injection of any number of zealots entails the *radicalisation* of some non-zealous users, turning them into zealots with the opposite opinion.

Our findings are illustrated through numerical simulations. Four different user graph topologies are considered: a fully-connected group where everyone is influenced by everyone else, an Erdős-Rényi random graph, a Barabási-Albert scale-free graph and a Watts-Strogatz small-world network. We find empirical averages to be close approximations of theoretical values. All code used for the simulations is available online.¹

2 Related Literature

Perhaps the earliest milestone in the study of opinion dynamics are the works from French [14] and Degroot [11] who studied how a society of individuals may or may not come to an agreement on some given topic. Assuming the society is connected and people repeatedly update their belief by taking weighted averages of those of their neighbours, they showed that consensus is reached. That is, everyone eventually agrees. Various other models have been developed since, to tackle the question of under which circumstances and how fast a population is able to reach consensus. Amongst others, [15] introduces immutable innate preferences, [3] studies the effect of homophily, [5] assumes individuals are perfectly rational and [22] accounts for the influence of external events.

The voter model was introduced independently by Clifford and Sudbury [9] and Holley [20] in the context of particles interaction. They proved that consensus is reached on the infinite \mathbb{Z}^d lattice. Several works have since looked at different network topologies, wondering whether consensus is reached, on which

¹ <https://github.com/antoinevendeville/howopinionscrystallise>.

opinion and at what speed. Complete graphs [19, 29, 31, 35], Erdős-Rényi random graphs [31, 35], scale-free random graphs [13, 31], and other various structures [31, 35] have been addressed. Variants where nodes deterministically update to the most common opinion amongst their neighbours have also been studied [7, 26].

An interesting case to consider is the one where zealots – i.e. stubborn agents who always keep the same opinion, are present in the graph. Such agents may for example represent lobbyists, politicians or activists, i.e. entities looking to lead rather than follow and who will not easily change side. One of those placed within the network can singlehandedly change the outcome of the process [24, 31]. If several of them are present on both sides, consensus is usually not reachable and instead opinions converge to a steady-state in which they fluctuate indefinitely [25, 34].

Recently, Mukhopadhyay [27] considered zealots with different degrees of zealotry and proved that time to reach consensus grows linearly with their number. They also showed that if one opinion is initially preferred—i.e. agents holding that opinion have a lesser probability of changing their mind—consensus is reached on the preferred opinion with a probability that converges to 1 as the network size increases. Klamser *et al.* [23] studied the impact of zealots on a dynamically evolving graph, and showed that the two main factors shaping their influence are their degrees and the dynamical rewiring probabilities.

With the increasing importance of social networks in the political debate and information diffusion, there has been a recent surge in research aiming at controlling opinions, often with the goal to reduce polarisation. In the context of the Voter and Friedkin-Johnsen models respectively, Yildiz *et al.* [34] and Goyal *et al.* [18] provide algorithms for selecting an optimal sets of stubborn nodes in order to push opinions in a chosen direction. Our work differs from the former in that we propose a more general strategy, that works on expectation for any connected graph. Yi and Patterson [33] formulate different constrained optimisation problems under the French-Degroot and the Friedkin-Johnsen models. They provide solutions in the form of optimal graph construction methods.

Still within the Friedkin-Johnsen paradigm, Chitra and Musco [8] prove that dynamically nudging edge weights in the user graph can reduce polarisation while preserving relevance of the content shown by the recommendation algorithm. Garimella *et al.* [16] propose a method to reduce polarisation through addition of edges in the network. The focus is put on which nodes to connect in order to get the best reduction in polarisation, while being sure that the edge is “accepted”—as extreme recommendations might not work because of the backfire effect. Finally, Cen and Shah [6] propose a data-driven procedure to moderate the gap between opinions influenced by a neutral or a personalised newsfeed. Importantly, they show that this can be done even without knowledge of the process through which opinions are derived from the newsfeed.

3 The Voter Model with Zealots

Consider a graph with n users labelled $1, \dots, n$ that can each hold opinion 0 or 1. Given an initial distribution of opinions, each user updates their opinion at the

times of an independent Poisson process of parameter 1 by adopting the opinion of one of its neighbours chosen uniformly at random. Letting $x_i(t)$ denote the opinion of user i at time t , we say that consensus is reached if almost surely all users eventually agree, i.e. if

$$\forall i, j, \quad \mathbb{P}(x_i(t) = x_j(t)) \xrightarrow[t \rightarrow \infty]{} 1. \quad (1)$$

On any finite connected network, consensus is reached [1]. Intuitively, no matter the current number of opinion-0 and opinion-1 users, there exists a succession of individual opinion changes with strictly positive probability that results in everyone holding the same opinion.

Here we place ourselves in the case where the user network is a connected graph with arbitrary degree distribution. This means there is a path from any user to another, and degrees of different users are not correlated with one another. Edges may be directed or not. We are interested in the particular situation where some of the agents are stubborn and never change their opinions. We call such agents *zealots*. Whenever a clock associated with a zealot rings, their opinion is not updated. They form an inflexible core of partisans within a group who bear great power of persuasion over the whole population. Both the position of these agents and initial opinions are assumed to be independent from the network topology.

We call 1-zealot a stubborn agent with opinion 1 and denote by z_1 their number. Similarly, z_0 will denote the quantity of 0-zealots. The remaining $n - z_0 - z_1$ users are free to change opinions during the whole duration of the process. If $z_0 > 0, z_1 = 0$ or $z_0 = 0, z_1 > 0$ then via similar arguments as for Eq. 1 consensus is reached on opinions 0 and 1 respectively. Here we are particularly interested in the case $z_0, z_1 > 0$, where there is always a strictly positive number of users with each opinion. This prevents consensus and instead the system reaches state of equilibrium in which it fluctuates indefinitely [25, 34]. We illustrate cases $\{z_0 > 0, z_1 = 0\}$ and $\{z_0, z_1 > 0\}$ in Fig. 1.

Our results are valid as long as at least one of z_0 and z_1 is strictly positive and we formally require $z_0 + z_1 > 0$. Importantly, we assume that for any fixed tuple (z_0, z_1) , the position of zealots is drawn uniformly at random. Letting Z be a random vector encoding these positions we formally write $Z \sim \mathcal{U}$.

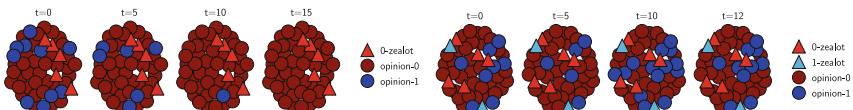


Fig. 1. (Left) Example realisation of the model on a complete graph at different times with $n = 50, n_1 = 10, z_0 = 5, z_1 = 0$. Because there are no 1-zealots, everyone eventually adopts opinion 0. **(Right)** Same setting except $z_1 = 2$. Because there are zealots in both camps, the system reaches a state of equilibrium where no opinion prevails.

4 Expected Opinion Diversity at Equilibrium

Assuming $z_0 + z_1 > 0$, $N_1(t)$ converges to a state of equilibrium, which is characterised by a time-independent stationary distribution π . If $z_0 = 0$ (resp. $z_1 = 0$) then π is the constant distribution δ_n (resp. δ_0). This equilibrium does not depend on the initial opinions of non-zealots but on the topology of the graph and the position of zealots. Let N_1^* be a random variable distributed under π . Its average value informs us on the limiting opinion diversity of the group. Authors of [25] proved that this average was $nz_1/(z_0 + z_1)$ for complete graphs in the $n \rightarrow \infty$ limit. We now show that this results holds on expectation for any connected graph where the position of zealots is drawn uniformly at random.

Theorem 1. *For any connected user graph and any z_0, z_1 such that $z_0 + z_1 > 0$, we have*

$$\mathbb{E}_{Z \sim \mathcal{U}} \mathbb{E}[N_1^*] = n \frac{z_1}{z_0 + z_1}. \quad (2)$$

This theorem states that the proportion of opinion-1 users is expected to endlessly fluctuate around the ratio $z_1/(z_0 + z_1)$. For example, having twice as many zealots as the other camp will on average lead to count twice as many partisans. Thus the camp that boasts the biggest quantity of zealots is expected to be of bigger size in the long run. The ratio $z_1/(z_0 + z_1)$ does not depend on n and can be interpreted as the average opinion diversity at equilibrium. Finally, if the graph is complete then all possible values of Z are equivalent and the left side of Eq. 2 is simply $\mathbb{E}[N_1^*]$.

Proof (Theorem 1). From [34, Theorem 2.1], the vector of individual opinions $(x_1(t), \dots, x_n(t))$ converges in distribution to a random vector (x_1^*, \dots, x_n^*) . From [34, Proposition 3.2] we have that $\mathbb{E}[x_i^*]$ is equal to the probability that a random walk on the user graph initiated at node i is absorbed by the set of 1-zealots. Because here we consider a connected graph with z_0, z_1 zealots placed uniformly at random, it holds that $\mathbb{E}_{Z \sim \mathcal{U}} \mathbb{E}[x_i^*] = z_1/(z_0 + z_1)$. There are $n - z_0 - z_1$ non-zealots and z_1 1-zealots, thus we have

$$\mathbb{E}_{Z \sim \mathcal{U}} \mathbb{E}[N_1^*] = z_1 + (n - z_0 - z_1) \mathbb{E}_{Z \sim \mathcal{U}} \mathbb{E}[x_i^*] \quad (3)$$

and Eq. (2) ensues.

4.1 Empirical Approximation

Let briefly discuss how to verify Theorem 1 through computer simulations. Assume we perform M simulations of our model on a given graph. Each time Z is drawn at random under \mathcal{U} at the beginning. Then the law of large numbers tell us that

$$\frac{1}{M} \sum_{m=1}^M \mathbb{E}[N_1^*] \xrightarrow{M \rightarrow \infty} n \frac{z_1}{z_0 + z_1}. \quad (4)$$

Now N_1^* is a random variable distributed under π . For a given realisation of $N_1(t)$, its expectation can be approximated empirically via

$$\frac{1}{T} \sum_{t \in \mathcal{T}} N_1(t) \xrightarrow{T \rightarrow \infty} \mathbb{E}[N_1^*] \quad (5)$$

where \mathcal{T} is a set of sufficiently large times and T denotes its size. Combining (4) and (5) we have:

$$\frac{1}{MT} \sum_{m=1}^M \sum_{t \in \mathcal{T}} N_1(t) \xrightarrow{M, T \rightarrow \infty} n \frac{z_1}{z_0 + z_1}. \quad (6)$$

Furthermore, we can also approximate the expected stationary distribution:

$$\frac{1}{MT} \sum_{m=1}^M \sum_{t \in \mathcal{T}} \mathbb{1}\{N_1(t) = k\} \xrightarrow{M, T \rightarrow \infty} \mathbb{E}_{Z \sim \mathcal{U}}[\pi_k]. \quad (7)$$

5 Control of Opinion Diversity

Theorem 1 is a useful tool for the control of opinions diversity at equilibrium. Let us represent a polarised online community by a connected graph of n individuals with a quantity $z_0 > 0$ of 0-zealots and no 1-zealot ($z_1 = 0$). For the sake of clarity we assume the graph to be complete but the strategy holds on expectation when $Z \sim \mathcal{U}$ in the more general case of any connected graph. The completeness assumption is also fairly reasonable for certain social platforms, such as Facebook or Reddit where everyone in a group or subreddit sees the posts of everyone else.

Because $z_0 > 0$ and $z_1 = 0$, the community is homogeneous: each member will end up adopting opinion 0 no matter what. The presence of such polarised groups hinder democratic debate online, as they prevent the evolution of opinion and reinforce pre-existing beliefs. To mitigate this phenomenon, we suggest injecting 1-zealots into the group. Doing so means forcing $z_1 > 0$ and thus at equilibrium, the group will be more diverse and divided between the two camps.

From Theorem 1, the average opinion at equilibrium is given by $z_1/(z_0 + z_1) \in [0, 1]$ and equals 0 here. Let us choose a target value $0 < \lambda < 1$, representing the level of diversity we want the group to reach. Typically it should be around 1/2 if the goal is to transform the echo chamber into a diverse sphere of opinions. It is immediate that injecting a quantity

$$z_1^* = \frac{\lambda}{1 - \lambda} z_0 \quad (8)$$

of 1-zealots yields $z_1^*/(z_0 + z_1^*) = \lambda$. In practice we round this value to the previous or next integer. If $\lambda = 1/2$, then we should add exactly as many 1-zealots as there are 0-zealots.

Note that the equilibrium opinion is only determined by the quantity of zealots on each side and other users do not affect it. The expected opinion held by such a user converges to $z_1/(z_0 + z_1)$ and thus their number and initial

opinions does not impact the equilibrium. Hence, one could “convert” users amongst the initial $n - z_0$ non-zealots into 1-zealots instead of adding external ones. That would lead to the same diversity, except that it comes with a limit on the quantity of 1-zealots because of the constraint $z_0 + z_1 \leq n$. Thus in that case, our strategy is only feasible if

$$z_0 + \frac{\lambda}{1 - \lambda} z_0 \leq n. \quad (9)$$

In a more general fashion, if we are limited in the number of zealots we can add to the network by z_1^{\max} , the optimal quantity of such agents is given by:

$$z_1^* = \min \left(z_1^{\max}, \frac{\lambda}{1 - \lambda} z_0 \right). \quad (10)$$

5.1 Backfire Effect

Numerous studies suggest that presenting certain users with opposing views might actually entrench them even deeper in their beliefs. This is known as the backfire effect. To account for it we study the scenario where in reaction to 1-zealots being created (either by addition or conversion), some of the non-zealous users will *radicalise* and become 0-zealots. Formally, we set that for each increment of z_1 , a quantity $\alpha < 1$ of non-zealous users become 0-zealots. The opinion at equilibrium is now given by

$$\frac{z_1}{(1 + \alpha z_1) z_0 + z_1} \quad (11)$$

and diversity λ is exactly reached with

$$z_1^* = \lambda z_0 D^{-1} \quad (12)$$

where $D := 1 - \lambda - \lambda \alpha z_0$. If $D > 0$ then $z_1^* > 0$ and we can inject this quantity of users into the network. If $D \leq 0$ however this becomes impossible as (12) is then either undefined or negative. In this case, we find that

$$z_1 \mapsto \left(\frac{z_1}{(1 + \alpha z_1) z_0 + z_1} - \lambda \right)^2 \quad (13)$$

is strictly positive and decreasing towards 0 over $\mathbb{R}_{>0}$. Thus the larger z_1 the closer we get to the target diversity, without ever reaching it. This means that the addition of 1-zealots entails the radicalisation of too many users into 0-zealots for it to ever counterbalance the spread of opinion 0. If we are limited in the number of zealots we can add to the network by z_1^{\max} , we have the natural optimal values for z_1 :

$$\begin{cases} z_1^* = \min(z_1^{\max}, \lambda z_0 D^{-1}) & \text{if } D > 0, \\ z_1^* = z_1^{\max} & \text{if } D \leq 0. \end{cases} \quad (14)$$

Note that in the case where we are converting existing users into zealots instead of injecting external ones, z_1^{\max} is given by the constraint

$$z_1^{\max} + \underbrace{(1 + \alpha z_1^{\max}) z_0}_{\substack{\text{updated } z_0 \\ \text{after backfire}}} = n. \quad (15)$$

6 Numerical Experiments

We now validate Theorem 1 via computer simulations. Let us set $(n, z_0, z_1) = (100, 20, 40)$ and $\phi = nz_1(z_0 + z_1) \simeq 66.7$. We consider four different undirected graph models with varying parameters:

- Complete graph with an edge between each pair of users,
- Erdős-Rényi random graph with density $p = 0.1, 0.3, 0.5$,
- Watts-Strogatz small-world graph with initial connections to the 4 nearest neighbours and rewiring probability $\omega = 0.01, 0.05, 0.1$,
- Barabási-Albert scale-free graph with $m = 1, 3, 5$ initial nodes and m new connections at each step.

For each model and each parameter we generate one connected user graph then perform $M = 500$ simulations. At the beginning of each, position of zealots and initial opinions are drawn uniformly at random, then we let the model evolve for 200 time units. In Fig. 2 we plot values of $N_1(t)$ averaged over all simulations for each graph model and parameter—up to $t = 50$ and not 200 for the sake of clarity. We observe a good correspondence with the theory, as after about 20 time units all the empirical averages seem to fluctuate closely around ϕ .

Insets show empirical distributions of N_1^* obtained via Eq. 7. For any given simulation, the empirical distribution is obtained by averaging all values of $N_1(t)$ starting at $t = 20$. This value was chosen by manual inspection, to make sure that the system had enough time to stabilise. Distributions for each choice of graph model and parameter are then averaged over all 500 simulations. We also show their means and its theoretical value ϕ . We observe very good correspondence between all the Erdős-Rényi and Watts-Strogatz graphs for all parameters, and a slightly worse one for the Barabási-Albert graphs. This might be due to the lower regularity in the graph topology, as when m increases, the distribution seems to get closer to the others. Finally, the means are all close to the theoretical value.

To quantify more precisely the discrepancy between theory and simulations, we look more closely at the difference between empirical $N_1(t)$ and ϕ . Again we only consider results from $t = 20$ onwards. For any graph model and parameter, the average relative error of the empirical means with respect to ϕ is calculated through:

$$\frac{1}{\phi} \frac{1}{M} \sum_{m=1}^M \left| \frac{1}{T} \sum_{t=20}^{100} N_1(t) - \phi \right| \quad (16)$$

where $M = 500$ is the number of simulations and T is the number of data points from $t = 20$ onwards. This value quantifies the average gap between empirical

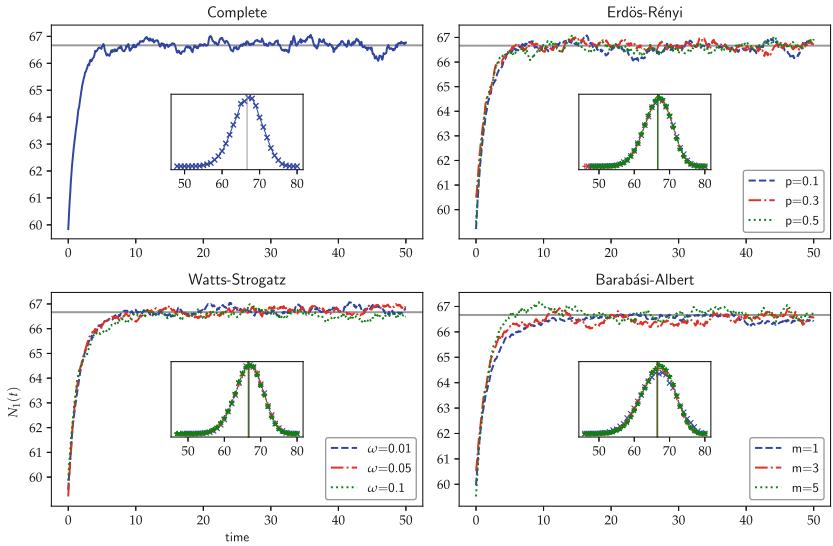


Fig. 2. Evolution of $N_1(t)$ for various graph models and parameters. Values are averaged over 500 simulations. Horizontal grey lines indicate the limiting expectation given by Theorem 1. Insets show empirical distributions at equilibrium.

means of $N_1(t)$ at equilibrium and their theoretical value ϕ , expressed as a percentage relative to the value of ϕ . Results are presented in Table 1 alongside standard deviations.

As to be expected the complete network boasts the most precise results, with an average relative error of $0.8\% \pm 0.6\%$ corresponding to a difference of 0.5 ± 0.4 users. Then in order of decreasing accuracy we have successively Erdős-Rényi, Watts-Strogatz and finally Barabási-Albert graphs. Each time the precision increases with the density of the graph, as the network gets closer and closer to the complete case. The worst case is the Barabási-Albert network with parameter $m = 1$, yielding an average relative error of $4.7\% \pm 3.5\%$ which corresponds to a difference of 3.1 ± 2.3 users.

Finally, we turn to the problem of controlling the diversity of opinions under the presence of a backfire effect. We consider a complete graph of $n = 100$ users with $z_0 > 0, z_1 = 0$ and three different target diversities $\lambda = 0.3, 0.5, 0.7$. The intensity of the backfire effect takes various values $\alpha \in \{0.025, 0.05, 0.075, 0.1\}$. The optimal quantity of 1-zealots is calculated according to (14), in the case where we are converting existing users and thus z_1^{\max} is given by Eq. 15. In Fig. 3 we plot z_1^* function of z_0 for each α . We cut the lines at the points where D drops below zero and reaching diversity λ exactly is not feasible anymore. In practice however, one could still convert z_1^{\max} users in order to increase diversity.

As expected, for all λ considered, the lower α the higher quantities 0-zealots it is possible to fight against. Peaks in the curve correspond to the tipping points where z_1^{\max} becomes smaller than the optimal value $\lambda z_0 D^{-1}$. Those peaks do not

Table 1. Relative errors between empirical averages of $N_1(t)$ at equilibrium and limiting expectation $nz_1/(z_0 + z_1)$, averaged over 500 simulations and with standard deviation.

Complete	$0.8\% \pm 0.6\%$	–	–
Erdős-Rényi	$p = 0.1$	$p = 0.3$	$p = 0.5$
	$1.7\% \pm 1.2\%$	$1.0\% \pm 0.7\%$	$0.9\% \pm 0.6\%$
Watts-Strogatz	$\omega = 0.01$	$\omega = 0.05$	$\omega = 0.1$
	$2.5\% \pm 1.9\%$	$2.6\% \pm 1.9\%$	$2.4\% \pm 1.9\%$
Barabási-Albert	$m = 1$	$m = 3$	$m = 5$
	$4.7\% \pm 3.5\%$	$3.3\% \pm 2.4\%$	$2.8\% \pm 2.1\%$

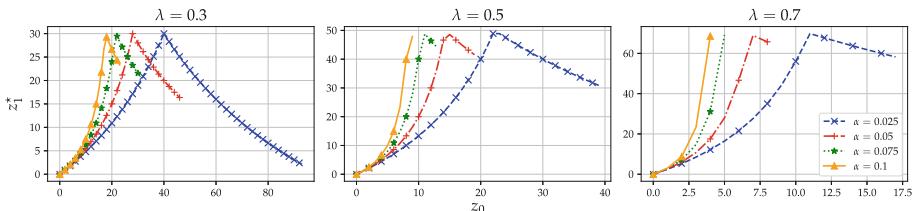


Fig. 3. Controlling diversity of opinions with backfire effect. Optimal z_1^* function of z_0 for $n = 100$, $\lambda = 0.3, 0.5, 0.7$ and various intensities of the backfire effect α . Note that axes scales differ from one plot to another.

appear in all cases: for the highest values of λ, α this point is never reached as z_0 becomes too high too quickly for diversity λ to be reached exactly. Additionally in Fig. 4 we plot the absolute difference between the diversity reached using $z_1^* = z_1^{\max}$ and λ in the cases where $D \leq 0$. The differences almost never drop below 10% in the cases considered, highlighting a non negligible gap between the realised diversity and the target.

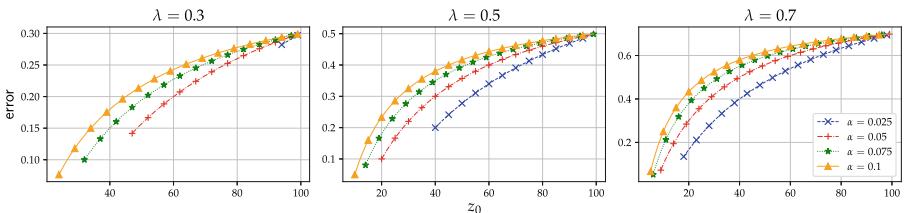


Fig. 4. Absolute difference between diversity reached with $z_1^* = z_1^{\max}$ and target λ in the cases where $D \leq 0$. Note that axes scales differ from one plot to another.

7 Conclusion and Future Work

In this paper we analysed the voter model with zealots on connected graphs with arbitrary degree distribution. Assuming that initial opinions and position of zealots are drawn uniformly at random, we extended existing results to provide closed-form expressions from (*i*) expected opinion distribution at equilibrium, and (*ii*) expected convergence time. We then used our findings to propose a simple method for the control of opinion diversity in a connected group of users that may or may not be subject to a backfire effect. Our analysis was supported through numerical simulations. Leads for further work include extensions such as considering more than two opinions or implementing variable degrees of zealotry.

Funding Information. This project was funded by the UK EPSRC grant EP/S022503/1 that supports the Centre for Doctoral Training in Cybersecurity delivered by UCL’s Departments of Computer Science, Security and Crime Science, and Science, Technology, Engineering and Public Policy.

References

1. Aldous, D.J., Fill, J.A.: Reversible markov chains and random walks on graphs (2002), unfinished monograph, recompiled (2014). <http://www.stat.berkeley.edu/~aldous/RWG/book.html>
2. Allington, D., Duffy, B., Wessely, S., Dhavan, N., Rubin, J.: Health-protective behaviour, social media usage and conspiracy belief during the COVID-19 public health emergency. *Psychol. Med.* **51**, 1–7 (2020)
3. Axelrod, R.: The dissemination of culture: a model with local convergence and global polarization. *J. Conflict. Resolut.* **41**(2), 203–226 (1997)
4. Bail, C.A., et al.: Exposure to opposing views on social media can increase political polarization. *PNAS* **115**(37), 9216–9221 (2018)
5. Banerjee, A., Fudenberg, D.: Word-of-mouth learning. *Games Econ. Behav.* **46**(1), 1–22 (2004)
6. Cen, S.H., Shah, D.: Regulating algorithmic filtering on social media (2020)
7. Chen, P., Redner, S.: Majority rule dynamics in finite dimensions. *Phys. Rev. E* **71**, 036101 (2005)
8. Chitra, U., Musco, C.: Analyzing the impact of filter bubbles on social network polarization. Proceedings of WSDM2020, pp. 115–123 (2020)
9. Clifford, P., Sudbury, A.: A model for spatial conflict. *Biometrika* **60**(3), 581–588 (1973)
10. Cota, W., Ferreira, S.C., Pastor-Satorras, R., Starnini, M.: Quantifying echo chamber effects in information spreading over political communication networks. *EPJ Data Sci.* **8**(1), 1–13 (2019). <https://doi.org/10.1140/epjds/s13688-019-0213-9>
11. DeGroot, M.H.: Reaching a consensus. *J. Am. Stat. Assoc.* **69**(345), 118–121 (1974)
12. Del Vicario, M., Zollo, F., Caldarelli, G., Scala, A., Quattrociocchi, W.: Mapping social dynamics on facebook: the brexit debate. *Soc. Netw.* **50**, 6–16 (2017)
13. Fernley, J., Ortgiese, M.: Voter models on subcritical inhomogeneous random graphs (2019)
14. French, J.R.: A formal theory of social power. *Psychol. Rev.* **63**, 181–94 (1956)
15. Friedkin, N.E., Johnsen, E.C.: Social influence and opinions. *J. Math. Sociol.* **15**(3–4), 193–206 (1990)

16. Garimella, K., De Francisci Morales, G., Gionis, A., Mathioudakis, M.: Reducing controversy by connecting opposing views. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM '17, pp. 81–90. Association for Computing Machinery, New York (2017)
17. Garimella, K., De Francisci Morales, G., Gionis, A., Mathioudakis, M.: Political discourse on social media: echo chambers, gatekeepers, and the price of bipartisanship. In: Proceedings of WWW 2018, pp. 913–922 (2018)
18. Goyal, M., Chatterjee, D., Karamchandani, N., Manjunath, D.: Maintaining ferment. In: 2019 IEEE 58th Conference on Decision and Control (CDC), pp. 5217–5222 (2019)
19. Hassin, Y., Peleg, D.: Distributed probabilistic polling and applications to proportionate agreement. *Inf. Comput.* **171**(2), 248–268 (2002)
20. Holley, R.A., Liggett, T.M.: Ergodic theorems for weakly interacting infinite systems and the voter model. *Ann. Probab.* **3**(4), 643–663 (1975)
21. Holone, H.: The filter bubble and its effect on online personal health information. *Croat. Med. J.* **57**, 298 (2016)
22. Jadbabaie, A., Molavi, P., Sandroni, A., Tahbaz-Salehi, A.: Non-bayesian social learning. *Games Econ. Behav.* **76**(1), 210–225 (2012)
23. Klamser, P.P., Wiedermann, M., Donges, J.F., Donner, R.V.: Zealotry effects on opinion dynamics in the adaptive voter model. *Phys. Rev. E* **96**, 052315 (2017)
24. Mobilia, M.: Does a single zealot affect an infinite group of voters? *Phys. Rev. Lett.* **91**, 028701 (2003)
25. Mobilia, M., Petersen, A., Redner, S.: On the role of zealotry in the voter model. *J. Stat. Mech. Theory Exp.* **2007**, P08029–P08029 (2007)
26. Mossel, E., Neeman, J., Tamuz, O.: Majority dynamics and aggregation of information in social networks. *Auton. Agents Multi-Agent Syst.* **28**(3), 408–429 (2013). <https://doi.org/10.1007/s10458-013-9230-4>
27. Mukhopadhyay, A.: Voter and majority dynamics with biased and stubborn agents. *J. Stat. Phys.* **181**, 1239–1265 (2020)
28. Pariser, E.: The Filter Bubble: What the Internet is Hiding from You. The Penguin Group, London (2011)
29. Perron, E., Vasudevan, D., Vojnovic, M.: Using three states for binary consensus on complete graphs. In: Proceedings of IEEE INFOCOM 2009, pp. 2527–2535 (2009)
30. Schaewitz, L., Krämer, N.C.: Combating disinformation: effects of timing and correction format on factual knowledge and personal beliefs. In: van Duijn, M., Preuss, M., Spaiser, V., Takes, F., Verberne, S. (eds.) MISDOOM 2020. LNCS, vol. 12259, pp. 233–245. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-61841-4_16
31. Sood, V., Tibor, A., Redner, S.: Voter models on heterogeneous networks. *Phys. Rev. E* **77**, 041121 (2008)
32. Williams, H.T., McMurray, J.R., Kurz, T., Hugo Lambert, F.: Network analysis reveals open forums and echo chambers in social media discussions of climate change. *Glob. Environ. Change* **32**, 126–138 (2015)
33. Yi, Y., Patterson, S.: Disagreement and polarization in two-party social networks (2019)
34. Yildiz, M.E., Ozdaglar, A., Acemoglu, D., Saberi, A., Scaglione, A.: Binary opinion dynamics with stubborn agents. *ACM Trans. Econ. Comput.* **1**(4), 1–30 (2013)
35. Yildiz, M.E., Pagliari, R., Ozdaglar, A., Scaglione, A.: Voting models in random networks. In: Proceedings of 2010 Information Theory and Applications Workshop (ITA), pp. 1–7 (2010)



Sensing Enhancement on Complex Networks

Markus Brede^(✉) and Guillermo Romero Moreno

School of Electronics and Computer Science, University of Southampton,
Southampton, UK
Markus.Brede@soton.ac.uk

Abstract. Previous work has shown that communication between agents with some preference towards adopting the majority opinion can enhance the quality of error-prone individual sensing from dynamic environments. In this paper, we compare the potential of different types of complex networks for such sensing enhancement. Numerical simulations on complex networks are complemented by a mean-field approach for limited connectivity that captures essential trends in dependencies. Our results show that whilst bestowing advantages on a small group of agents degree heterogeneity tends to impede overall sensing enhancement, while clustering and spatial structure play a more nuanced role depending on overall connectivity. We find that for low connectivity sensing enhancement is maximised by random regular networks, whereas for large connectivity best sensing enhancement is found for ring graphs.

Keywords: Complex networks · Opinion dynamics · Sensing enhancement

1 Introduction

Sensing and processing information about uncertain environments is important for survival in many types of collectives in the animal world [1, 3] as well as in human populations [20]. Refining information about a complex dynamic environment is particularly important if individuals' sensing abilities are limited or information is highly complex and difficult to evaluate. In principle then, the quality of available information could be improved by pooling multiple individual estimates. However, in many examples in the animal world, such as house-hunting insects [4] or honeybees [2], collective vigilance in fish [5], and obviously in human populations this can also be achieved by information sharing between members of the population.

Given the increasing use of social media for news consumption and sourcing political information [6, 7], in the human context, refining information through peer communication might be of particularly relevance. Importantly, peer communication on social media often happens through social networks, and hence gaining insights about the impact of network structure on information refinement is a topical research question [19, 21].

Inspired by social insects and applications in swarm robotics, collective decision-making has found much attention in the past [22] and some previous studies have argued that optimal decision making requires systems to be at criticality, i.e. operating at the boundary between order and chaos [9]. However, other recent work has also shown that agents can enhance the quality of sensed information through communication in a social network close to the bifurcation point of a bi-stable system [12, 13]. Similar mechanisms to [12, 13] have also found attention before in swarm robotics [8], but in the latter context, instead of directly modelling limited sensing abilities of individuals, models have typically included a positive feedback effect related to the quality information in problems related to the best-of- n problem [10, 11].

Whereas the mechanism of [12, 13] has been evaluated in detail in [13] for all-to-all connected systems and in [12] on spatial networks, a more general comparison of the role of network topology is still outstanding and we address this issue in this paper. For this purpose, we first develop a simplified version of the model of [13], which ignores details in the modelling of signal production and signal reception and focuses on the mechanism of information refinement in the bi-stable system. Analysing our modified model, we develop a mean-field approach for limited connectivity, and provide results about the potential for information refinement of different complex network topologies.

2 Model and Mean-Field Results

Consider a population of N agents (or sensors) identified with nodes of a social network given by an adjacency matrix $A = \{a_{ij}\}_{i,j=1}^N$ with $a_{ij} = 1$ if nodes i and j are connected and $a_{ij} = 0$ otherwise. We further presume that at time t an agent i is characterised by a binary state $s_i(t) \in \{0, 1\}$, where, without loss of generality, we presume that a state $s = 1$ represents that the agent has knowledge of the correct state of the environment. We also consider agents that are equipped with imperfect sensors with which they can determine the correct state of the environment with probability q (and receive incorrect information with probability $1 - q$), where $1/2 < q < 1$. The value of q represents an agent's sensing accuracy, where $q = 0.5$ implies random sensing and $q > 0.5$ models sensing of the correct state of the environment with better than equal chance. In the paper, we do not calibrate the model to a real-world scenario, but rather explore outcomes over ranges of parameters. Attempting to obtain correct information about the state of the environment, agents can sample social information about the states of their network neighbours or can use their own (inaccurate) sensors to directly measure the state of the environment. In more detail, we presume that updating happens in discrete time and in synchronous, parallel updates, such that at each time step t every agent measures the state of the environment with probability p or samples social information from its network neighbours with probability $1 - p$. Below, we shall refer to p as an agent's sensing intensity. Furthermore, the state of the environment is also assumed to be binary and changes dynamically at rate u at each update. Specifically, this implies that all

agents switch states from s_i to $1 - s_i$ when the environment changes, i.e. since the environment is binary, after a change agents that had the correct information now have incorrect information, and agents with incorrect information have correct information. When sampling social information, an agent i will adopt the state corresponding to correct information about the environment with probability

$$Pr\{s_i(t+1) = 1\} = \frac{(1/k_i \sum_j a_{ij} s_j(t))^\alpha}{(1/k_i \sum_j a_{ij} s_j(t))^\alpha + (1 - 1/k_i \sum_j a_{ij} s_j(t))^\alpha}, \quad (1)$$

where $k_i = \sum_j a_{ij}$ is the in-degree of node i and the exponent α measures the selection strength with which i favours the majority state in its neighbourhood. The parameter α could be interpreted as the strength of complex contagion in non-linear voter-model-like spreading dynamics as in [14], and corresponds to the selection parameter β in the slightly different model formulation of [13].

Let us label the probability that agent i is in possession of the correct information at iteration t by $P_i(t)$. In a mean-field description, the dynamics of P_i is thus given by

$$P_i(t+1) = pq + (1-p) \begin{cases} \frac{(1/k_i \sum_j a_{ij} P_j(t))^\alpha}{(1/k_i \sum_j a_{ij} P_j(t))^\alpha + (1 - 1/k_i \sum_j a_{ij} P_j(t))^\alpha} & \text{with prob. } 1-u \\ \frac{(1 - 1/k_i \sum_j a_{ij} P_j(t))^\alpha}{(1/k_i \sum_j a_{ij} P_j(t))^\alpha + (1 - 1/k_i \sum_j a_{ij} P_j(t))^\alpha} & \text{otherwise,} \end{cases} \quad (2)$$

where the choices in Eq. (2) correspond to scenarios in which the signal from the environment has switched or not. We see that the dynamics of P are determined by the four parameters: (i) the intensity of sensing p , (ii) the accuracy of sensing q , (iii) the rate of change of the environment u , and (iv) the selection strength α .

To proceed, let us first examine the outcome of the updating dynamics in the absence of change in the environment. In this case, using a mean-field description and examining the stationary outcome P , one obtains

$$P = pq + (1-p) \frac{P^\alpha}{P^\alpha + (1-P)^\alpha}, \quad (3)$$

a system that exhibits a transition from a bi-stable phase for small p in which the system gets locked into either a majority $s = 0$ or $s = 1$ state and another phase for larger p in which most agents will align with the correct signal $s = 1$. This is apparent from the corresponding orbit diagram for $\alpha = 2$ and $q = 0.51$, as illustrated in panel (a) of Fig. 1 with dashed lines, revealing a bifurcation point at $p_c \approx 0.458$. Assuming a static environment, we thus see that in the bi-stable phase agents can get “lucky” and accidentally get locked into a majority correct state (provided there is an initial bias towards more agents in the correct state). In this first phase agents cannot follow a fluctuating environment and for any $u > 0$ initial luck will average out such that agents will only be in the correct state with roughly probability $1/2$ whenever their locked-in state happens to coincide with the correct environmental state. In contrast, in the second phase, a typically smaller proportion of agents will adopt the correct state, however

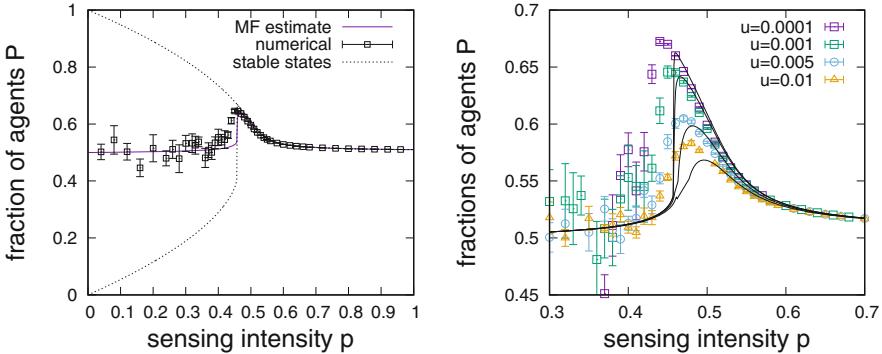


Fig. 1. (a) Dependence of the average fraction of correctly sensing agents P on the sensing intensity p . The figure compares numerical data obtained for an all-to-all connected system with 1000 nodes (black squares) to the mean-field orbit diagram for the bi-stable system and a mean-field estimate for $P(p)$ from Eq. (3) (dotted lines) along with an estimate of the stationary outcome of the switching dynamics based on Eq. (5) (magenta line). Parameters are switching rate $u = 0.001$, $\alpha = 2$, and sensing accuracy $q = 0.51$. Numerical data are from simulations over 10000 iterations of the dynamical process averaged over 10 independent runs. In the orbit diagram we find a critical point $p_c \approx 0.458$ such that below p_c the system is bi-stable and above p_c it follows the sensing of all agents and manages to adapt to changing signals. For a switching rate of the external signal of $u = 0.001$, the maximum fraction of agents aware of the correct signal is found at $p \approx 0.466$, slightly above the bifurcation point. (b) Comparison of the mean-field estimate (based on Eq. (5), black lines) vs. numerical data for different switching rates u .

independent of initial conditions. In this phase, the intensity of sensing is large enough such that agents can follow the external signal. However, the more often agents sense —i.e. the larger p — the stronger their limitation to their sensor accuracy, which can only sense the external signal correctly with probability q . Panel (a) of Fig. 1 makes it plain that the consensus state of the population can achieve states P in which the consensus perception of the external signal is considerably more accurate than an agent's own sensing ability q , which is clearly apparent close to the bifurcation point around $p = 0.458$. These results are also supported by numerical simulations (square symbols in Fig. 1a) of a system of $N = 1000$ all-to-all coupled agents for $\alpha = 2$ and $u = 0.001$, where we find the point of maximum sensing enhancement $p_{max} = 0.466$ with $P \approx 0.63$ (i.e. considerably larger than individual sensing accuracies of $q = 0.51$), i.e. just slightly to the right of the bifurcation point at $p_c = 0.458$.

Larger rates of change of the environment require faster responses, such that the requirement for as large as possible an equilibrium consensus state is balanced against the ability to actually reach that state before system change. As a consequence, for larger u the point of maximum sensing enhancement shifts away from the bifurcation point to the right, and maximum achievable correct sensing declines, as seen in panel (b) of Fig. 1. The above represents a speed-accuracy trade-off [15].

To obtain an analytical mean-field estimate for the dependence of P on u , we make another simplifying assumption and consider that switches in the state of the environment occur deterministically every $T = 1/u$ iterations. The above approximation ignores the exponential distribution of times of constant environments. An estimate of the dependence of P on u can then be obtained from averaging over one period of the (stationary) mean-field dynamics

$$P(t+1) = pq + (1-p) \begin{cases} P(t)^\alpha / (P(t)^\alpha + (1 - P(t))^\alpha) & \text{for } 0 \leq t < T \\ (1 - P(t))^\alpha / (P(t)^\alpha + (1 - P(t))^\alpha) & \text{for } t = T \end{cases} \quad (4)$$

and

$$P = \frac{1}{T} \sum_{t=0}^T P(t). \quad (5)$$

Results based on Eq. (5) are included along with numerical simulations in Fig. 1 (solid lines in both panels) and give a reasonable match to numerical data for the all-to-all connected system, but always slightly underestimate optimal sensing enhancement.

We further note that whilst the mean-field description above does not explicitly model sparse connectivities, numerical results (discussed later in Fig. 4) show a clear dependence of optimal sensing points on degree. To include this effect, we develop a mean-field approximation for finite degrees, which essentially factor in through exact configurations of states in an agents neighbourhood and can be accommodated by modifying (3) by

$$P(t+1) = pq + (1-p) \sum_{j=0}^k \binom{k}{j} P(t)^j (1 - P(t))^{k-j} \frac{(j/k)^\alpha}{(j/k)^\alpha + (1 - j/k)^\alpha}, \quad (6)$$

which assumes that all nodes of the graph have roughly equal degrees and j represents the number of neighbours in state $s(t) = 1$. Equation (6) can then be used to adapt Eq. (4) for limited degrees and can be inserted into Eq. (5) to yield estimates for the fraction of agents in possession of correct information. Estimates based on Eq. (6) are compared to numerical data in panel (a) of Fig. 2, where we see that whereas there are considerable differences between mean-field estimates and numerical data for small connectivity, reasonable agreement is found for connectivities above around $\langle k \rangle = 40$, and the main trends of the dependencies in the numerical data are captured by the limited connectivity mean-field approach (dotted lines). This contrasts with the all-to-all-coupling based mean-field approach (solid line), which fails to account for the observed limited range of sensing accuracies q for which consensus enhancement is possible.

3 Numerical Results for Complex Networks

In this section we detail numerical results to explore the role of complex networks for sensing enhancement. For these experiments, we focus on four different networks topologies: (i) random regular graphs (RRG), (ii) Erdös–Rényi

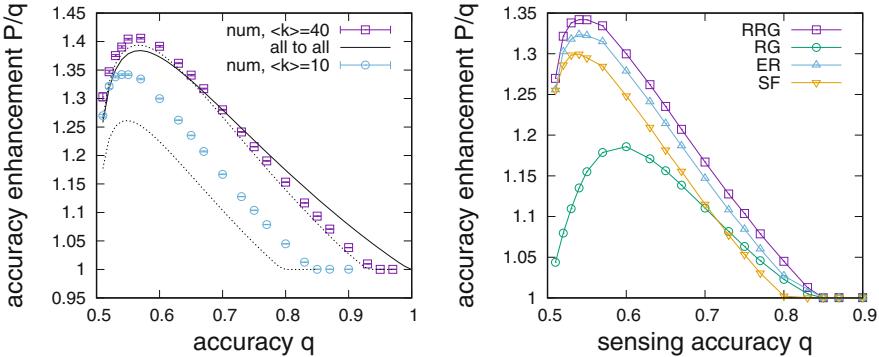


Fig. 2. Maximum sensing enhancement P/q as a function of the sensing accuracy q . (a) Comparison between numerical data for random regular graphs with connectivity $\langle k \rangle = 10$ and $\langle k \rangle = 40$ with the mean–field estimates for limited connectivity (dotted lines) and all-to-all coupling (solid line). (b) Comparison for different types of complex networks for $u = 0.001$, $\alpha = 2$, $\langle k \rangle = 10$.

random graphs (ER), (iii) Barabási–Albert scale-free networks (BA), and (iv) ring graphs (RG). These networks have been selected to explore the role of degree heterogeneity (by comparing networks (i)–(iii)) and exploring the role of spatial embedding (by comparing (i) and (iv)). For all of these networks, we are interested in the maximum sensing enhancement achievable when tuning the sensing intensity. To determine these, we typically run numerical experiments for a range of sensing intensities p with between 10^4 and 10^5 updates of the switching dynamics (depending on u) to find $P_{\max}(p)$ for each individual network, and then average over multiple network realisations to determine an estimate for the average maximum sensing enhancement for the ensemble. As the numerics are relatively costly, experiments presented below are typically evaluated for networks of size $N = 1000$ and averages are calculated over 20 network realisations. When exploring parameter dependencies, we typically focus on a low sensing accuracy $q = 0.51$ to investigate sensing enhancement that can be achieved in very adverse settings.

In a first experiment we evaluate the relative maximum sensing enhancement P/q of different network topologies as a function of the sensing accuracy q . For this purpose, panel (b) of Fig. 2 shows the dependence of P/q as a function of q for networks with $\langle k \rangle = 10$ for $u = 0.001$ and $\alpha = 2$. We note that the possible sensing enhancement as a function of q typically has a maximum for low q allowing for up to 30% improvement of sensing accuracy depending on network topology. Additionally, depending on network structure, sensing enhancement is only possible for a limited range of sensing accuracies and there typically exists a sensing accuracy q_c beyond which sensing enhancement is no longer possible. Comparing the four different network topologies, we find largest sensing enhancement for RRG's at around $P_{\max}/q \approx 1.35 \pm 0.01$, somewhat lower enhancement for ER with $P_{\max}/q \approx 1.33 \pm 0.01$ and SF networks with $P_{\max}/q \approx 1.3 \pm 0.01$

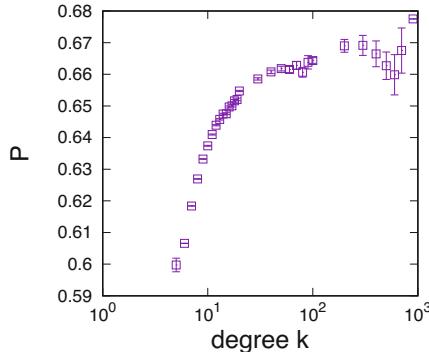


Fig. 3. Average probability to sense the correct state of the environment as a function of degree averaged over 10 SF networks with $N = 10^4$ for $u = 0.001$, $\alpha = 2$, $q = 0.51$, $p = 0.31$ and $\langle k \rangle = 10$.

and lowest for RG's with $P_{max}/q \approx 1.17 \pm 0.01$. The corresponding enhancement cut-offs are $q_c \approx 0.83 \pm 0.01$ for RRG, ER, and RG, and $q_c \approx 0.8 \pm 0.01$ for SF networks.

We thus see that the structure of the social network connecting the agents can have a strong impact on achievable sensing enhancement. Results presented in Fig. 2b suggest that in otherwise random networks heterogeneity plays a major role, seemingly impeding sensing enhancement. However, degree-regularity does not seem to guarantee best performance, which is seen in the worse characteristics for ring graphs compared to RRGs.

To further explore the role of degree heterogeneity, it proves instructive to investigate the dependence of the probability to accurately sense the environment on node degree, which is plotted in Fig. 3. As one would expect, we see that hub nodes tend to have better awareness of the environment than low-degree nodes, but, after an initial steep increase, increases tend to quickly saturate with degree from $k \approx 20$ on. We thus see a reason for the poorer performance of the SF networks: in their case connectivity spent on hub nodes only gives a small improvement in sensing which comes at the cost of a relatively larger deterioration of correct information in low-degree nodes.

To proceed, Fig. 4 compares the dependence of optimal sensing on the connectivity of networks and mean-field estimates based on Eq. (6), both in terms of the optimal sensing enhancement in Fig. 4a and for the sensing intensity p required at the optimal point in panel Fig. 4b. These results lead to a number of observations. First, close inspection shows that larger connectivity does not always lead to improved sensing. In contrast, there exists a connectivity at which sensing is maximally enhanced at around $\langle k \rangle = 20$ for RRG, ER, and SF and at roughly $\langle k \rangle = 30$ for RGs. The existence of such a connectivity maximum is also reproduced in the mean-field estimate which fits reasonably well with results for RRG, ER and SF networks for which differences become very small for large connectivities. Nevertheless, for low connectivity, $\langle k \rangle < 20$ we note that gener-

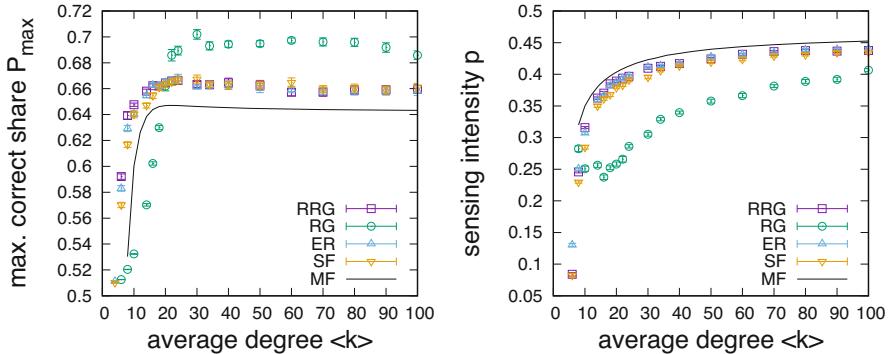


Fig. 4. Dependence of the point of optimal sensing enhancement on connectivity for different networks. (a) Optimal sensing P_{max} vs. connectivity $\langle k \rangle$. (b) Required sensing intensity at the optimum vs. connectivity. Numerical data obtained from simulations of 10^4 iterations of the updating process and averaged over 20 networks of size $N = 1000$ for $q = 0.51$, $\alpha = 2$, and $u = 0.001$. The black lines give mean-field estimates.

ally RRGs perform better than ER which are in turn superior to SF networks and RGs which perform significantly poorer in terms of sensing enhancement. Differences between these networks tend to become larger the lower the connectivity of the graph. For $\langle k \rangle > 20$ the order of performance is maintained for RRG, ER, and SF networks, but it is apparent that for large connectivities RGs have a clear advantage. This advantage manifests itself in terms of clearly superior performance, but also in terms of much reduced sensing intensities required at the optimal point, see Fig. 4b. For the setting investigated in Fig. 4 we also note that sensing enhancement is only possible for $\langle k \rangle \geq 6$, otherwise no sensing enhancement is found.

As a last parameter dependency of interest, Fig. 5 shows numerical results for the effect of the selection strength α for networks with $\langle k \rangle = 40$ (where $\langle k \rangle = 40$ has been chosen as for this connectivity the sensing enhancement for all network types has roughly saturated with degree). As for the dependence on connectivity (see Fig. 5a), we again note the existence of an optimal α at which the largest sensing enhancement is possible. For RRG, ER, and SF networks, optimal α is found at around $\alpha \approx 1.5$ (which roughly coincides with the mean-field prediction); for RGs the optimal α is located at slightly larger $\alpha \approx 1.7$. From the α -dependencies we also see that the order of network performance is consistently maintained, i.e. for $\langle k \rangle = 40$ RGs allow for more enhancement than RRGs, ERs, and SF networks, differences between which become small. As a last observation from Fig. 5a, we also see that sensing enhancement is only possible for $\alpha > 1$, i.e. only in the presence of complex contagion and not in linear voter-like dynamics with $\alpha = 1$ [16].

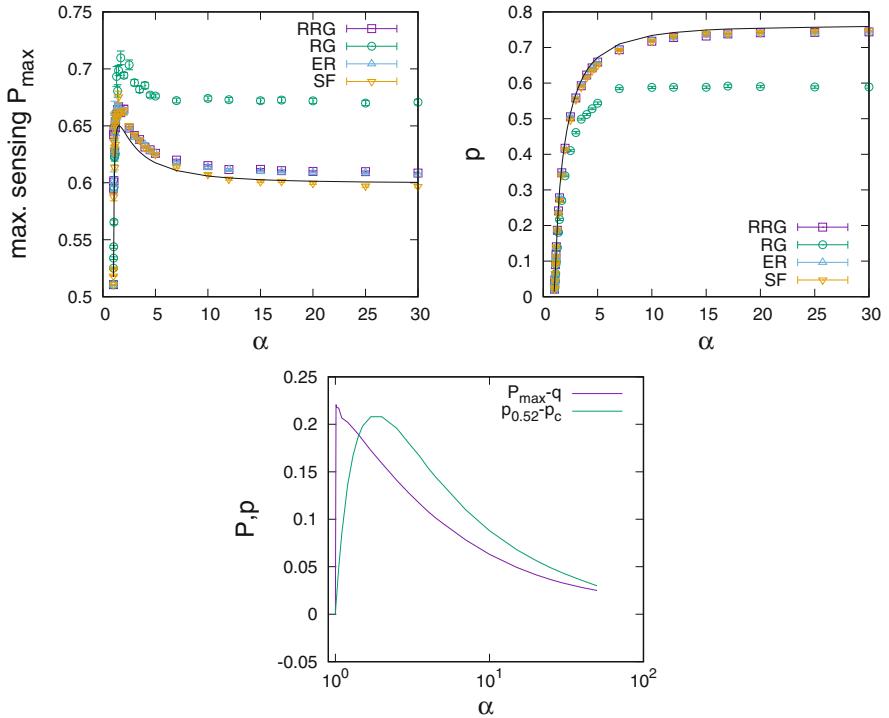


Fig. 5. Dependence of the point of maximum sensing enhancement on α . (a) Dependence of optimal sensing P_{\max} on α for various networks with $\langle k \rangle = 40$, $u = 0.001$, and $q = 0.51$. One notes there is a max value of α , such that sensing is maximally enhanced. (b) Dependence of the required sensing intensity p at the maximum point on α for various complex networks. The solid line gives the mean-field estimate. (c) Results from the mean-field analysis for the dependence of P_{\max} and the width of the enhancement region $p_{P>0.52}$ on α . More consensus enhancement is possible for smaller $\alpha > 1$, but the width of the peak converges to zero and transients become longer the closer one gets to $\alpha = 1$ from above.

Figure 5b continues the analysis by showing the dependence of the required sensing intensity at the optimal point on α . We again note that differences between RRGs, ERs and SF networks are rather small and close to the mean-field expectation. In contrast, and in particular for larger α , sensing intensities required at RGs are noticeably smaller.

Finally, results from an analysis of the finite-connectivity mean-field orbit diagram are presented in Fig. 5c. In more detail, we plot the sensing enhancement at the bifurcation point as a function of α . One notes that the closer α to $\alpha = 1$ from above, the larger the possible sensing enhancement, which would suggest an optimum at the smallest possible α approaching $\alpha = 1$. However, the smaller α , the slower the response dynamics after a switch in the environmental signal. Hence, as argued earlier, points of optimal sensing will generally be

found at slightly larger values than the sensing intensity at the bifurcation point. Values will be the larger, the shorter the time-scale of environmental change. Figure 5c also shows that the size of the parameter region of sensing intensities for which meaningful sensing enhancement is possible quickly converges to zero as α approaches one. This in effect limits possible sensing enhancement for small α and explains the existence of an optimal value of α distinctly larger than $\alpha = 1$ —again an effect related to speed-accuracy trade-offs [15].

4 Conclusions

In this paper we have presented a detailed investigation of the potential for information refinement of different complex network topologies. Results indicate that there are differences between different network structures, with both degree heterogeneity and spatial embedding (or local cohesiveness) affecting information refinement capabilities. Our study has demonstrated that, for low connectivities, information refinement is enhanced by reduced degree heterogeneity, but impeded by spatial structure, such that superior refinement can be found in regular random graphs. This finding is akin to studies of optimal synchronization [17] which have found that the stability of synchronized states is maximised in a particular class of regular networks. For large connectivity, spatial embedding is found to give benefits to information refinement and degree heterogeneity only plays a marginal role.

Further to the exploration of the role of network characteristics on sensing enhancement, we have also seen that the selection strength exponent α when sampling neighbour states plays an important role. First, our results indicate that optimal enhancement will typically be found for values of α within the $1 < \alpha < 2$ range, indicating that the majority-rule often considered in robotics applications [11] might not necessarily be the best choice. Second, we have seen that sensing enhancement is only possible for $\alpha > 1$. The result may point towards an evolutionary origin of complex contagion in opinion dynamics observed in human populations [18] related to filtering information from peers.

Whereas our paper has explored the role of some network properties for sensing enhancement, our study has been restricted to undirected networks and was mainly focused on degree heterogeneity and spatial embedding. Questions about the role of other network characteristics, such as degree mixing, average distances, motifs, or differences in in- and out-degree heterogeneity might be interesting topics for future research.

Acknowledgements. The authors acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work. MB acknowledges support from the Alan Turing Institute (EPSRC grant EP/N510129/1) and the Royal Society (grant IES R2 192206).

References

1. Dall, S.R., Giraldeau, L.A., Olsson, O., McNamara, J.M., Stephens, D.W.: Information and its use by animals in evolutionary ecology. *Trends Ecol. Evol.* **20**, 187–193 (2005)
2. Seeley, T.D.: *Honeybee Democracy*. Princeton University Press, Princeton (2010)
3. Sumpter, D.J.: *Collective Animal Behavior*. Princeton University Press, Princeton (2010)
4. Franks, N.R., Pratt, S.C., Mallon, E.B., Britton, N.F., Sumpter, D.J.T.: Information flow, opinion polling and collective intelligence in house-hunting social insects. *Phil. Trans. Roy. Soc. B* **357**, 1567–1583 (2002)
5. Ward, A.J., Herbert-Read, J.E., Sumpter, D.J., Krause, J.: Fast and accurate decisions through collective vigilance in fish shoals. *Proc. Natl. Acad. Sci.* **2001**, 007102 (2011)
6. Gottfried, J., Shearer, E.: News use across social media platforms 2016. Pew research centre (2016). <http://www.journalism.org>
7. de Z'uniga, H.G., Weeks, B., Ard'evol-Abreu, A.: Effects of the news-finds-me perception in communication: social media use implications for news seeking and learning about politics. *J. Comput.-Mediat. Commun.* **22**, 105–123 (2017)
8. Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M.: Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell.* **7**, 1–41 (2013)
9. Hidalgo, J., Grill, J., Suweis, S., Muñoz, M.A., Banavar, J.R., Maritan, A.: Information-based fitness and the emergence of criticality in living systems. *PNAS* **111**, 10095–10100 (2014)
10. Prasetyo, J., De Masi, G., Ferrante, E.: Collective decision making in dynamic environments. *Swarm Intell.* **13**, 217–243 (2019)
11. Valentini, G., Ferrante, E., Hamann, H., Dorigo, M.: Collective decision with 100 Kilobots: speed versus accuracy in binary discrimination problems. *Auton. Agent Multi-Agent Syst.* **30**, 553–580 (2016)
12. Salahshour, M., Rouhani, S., Roudi, Y.: Phase transitions and asymmetry between signal comprehension and production in biological communication. *Sci. Rep.* **9**, 3428 (2019)
13. Salahshour, M.: Phase diagram and optimal information use in a collective sensing system. *Phys. Rev. Lett.* **123**, 068101 (2019)
14. Alshamsi, A., Pinheiro, F.L., Hidalgo, C.A.: Optimal diversification strategies in the networks of related products and of related research areas. *Nat. Commun.* **9**, 1328 (2018)
15. Franks, N.R., Dornhaus, A., Fitzsimmons, J.P., Stevens, M.: Speed versus accuracy in collective decision making. *Proc. Roy. Soc. B* **270**, 2457–2463 (2003)
16. Holley, R., Liggett, T.: Ergodic theorems for weakly interacting infinite systems and the voter model. *Ann. Probab.* **3**, 643–663 (1975)
17. Donetti, L., Hurtado, P.I., Muñoz, M.A.: Entangled networks, synchronization, and optimal network topology. *Phys. Rev. Lett.* **95**, 188701 (2005)
18. Centola, D.: The spread of behavior in an online social network experiment. *Science* **329**, 1194–1197 (2010)
19. Talamali, M.S., Saha, A., Marshall, J.A.R., Reina, A.: When less is more: robot swarms adapt better to changes with constrained communication. *Sci. Robot.* **6**, eabf1416 (2021)
20. Surowiecki, J.: *The Wisdom of Crowds: Why the Many Are Smarter than the Few and How Collective Wisdom Shapes Business, Economies, Societies, and Nations*. Doubleday Books, New York (2004)

21. Becker, J., Brackbill, D., Centola, D.: Network dynamics of social influence in the wisdom of crowds. *Proc. Natl. Acad. Sci. United States Am.* **114**(26), E5070–E5076 (2017)
22. Bose, T., Reina, A., Marshall, J.A.: Collective decision-making. *Curr. Opin. Behav. Sci.* **16**, 30–34 (2017)



Cascading Failures and the Robustness of Cooperation in a Unified Scale-Free Network Model

Mingxuan He^(✉), Matthew Gao, Yang Gao, and Fernanda M. Elliott

ELBICA Laboratory, Grinnell College, 1116 8th Ave, Grinnell, IA 50112, USA
hemingxu@grinnell.edu
<https://elbica.cs.grinnell.edu/start-here/>

Abstract. In this paper, we examine the effects the cascading failures phenomenon has on unified scale-free complex networks utilizing prisoner’s dilemma game (PDG) dynamics. We find that a single defector may severely impact a large network that may result in a total failure of the entire network. We extend existing results to a network model that unifies the scale-free property and the high-clustering property. Furthermore, we observe that highly connected networks are more vulnerable to the cascading failure effect than less connected ones, as exhibited in both lower average survival rate and lower extinction boundary across different network topologies. As we attempt to examine the cascading failure effect in a more realistic network that is scale-free and highly clustered, we believe that these findings may be beneficial to studying such effects in the real world.

Keywords: Klemm-Eguiluz networks · Evolutionary games · Cascading failure effects · Prisoner’s Dilemma

1 Introduction

During a pandemic, we see individuals often face conflicts between their own interests and the community’s collective interest. Is it more tempting to go out and without masks to enjoy personal freedom from a minor inconvenience, or is it better to follow quarantine and put on masks to maintain community health and prevent the spread of disease? When such dilemmas occur, cooperation is key to the overall welfare of the group. In recent years, evolutionary game theory has provided powerful tools to model the emergence of cooperation and defection, especially in structured populations where agents only interact with a certain subset of the population [2, 8, 9].

One major challenge in studying cooperation in networked games is the difficulty to generalize most results due to the complex and sometimes chaotic nature of these games. The robustness of cooperation reacts to almost any modification in the model, and the reaction is often drastic [10]. This has led to a wide range

M. Gao and Y. Gao—Equal contribution.

of empirical and analytical conclusions in literature about what factors inhibit or promote cooperation. After the first seminal work by Nowak and May [8] which found enhanced cooperation in PDG played on a lattice network, Hauert *et al.* [2] showed that population structure in fact often inhibits cooperation for the Snowdrift game on the same lattice network. After the work is extended to complex networks, even more diverse and sometime contradictory results were published [13]. Santos *et al.* drew conclusions from hubs in scale-free networks to argue that network topology plays a determinant role in networked games, and therefore proposing the scale-free model as the unifying network model for the emergence of cooperation [11, 12]. However, an extensive survey by Roca *et al.* [10] suggested that population structure has no general effect on evolutionary outcomes, and instead emphasized on the role of clustering, degree heterogeneity, community structure, and the update rule for agents' strategies. Meanwhile, other network statistics including average degree and degree assortativity were also found relevant in a variety of networked games, e.g. [6, 9, 16]. While most of these works study dynamics on populations of fixed size and structure, which differ from the experiments we ran here with a dynamic population with a failure mechanism, they nonetheless provide insights for choosing potential factors for us to investigate, the details of which we cover in the Experiments section.

A failure mechanism was introduced to networked games in order to model the collapse of financial and economic networks during the 2008 financial crisis, when the initial collapse of one or few large financial institutions triggered a worldwide recession [17]. Their results showed that a small number (even one) of initial defectors can cause the removal of a large proportion of the population, and cooperators are the sole survivors.

Our work extends and furthermore complements the results from [17] by adopting as our population structure a special growing model for scale-free networks, the Klemm-Eguíluz (KE) model, which unifies key concepts from both small-world models and scale-free models [3]. By tuning a mixing parameter μ and a connectivity parameter m to generate various types of structured and unstructured scale-free networks, we are able to make general observations on network characteristics that affect the population's robustness against cascading failure caused by the spread of defection from a single defector.

Our main findings are (1) with a high connectivity parameter m for the Klemm-Eguíluz model, the average survival rate of the agents in network decreases to zero. In other words, a graph that is closer to a fully connected graph would have less robustness according to the parameters ran in our experiments. This finding is independent of the mixing parameter μ . Meanwhile, (2) with a low connectivity parameter, the extinction boundary α^* decreases faster for topologies with higher mixing parameters.

This paper is organized as follows. Section 2 explains the dynamics of networked evolutionary PDG. Section 3 details the Klemm-Eguíluz model and relevant parameters. Section 4 presents experimental results and qualitative explanations. Finally, Sect. 5 summarizes our results and poses future research questions regarding networks with different parameters and dynamics.

2 Evolutionary Game Dynamics

In game theory, the prisoner’s dilemma game (PDG), originally formalized by Albert W. Tucker who presented the game through hypothetical prison sentences [5], is one of the most recognized interaction mechanisms in the studies of multi-agent dynamics. To study agents’ self-organizing behaviors in the presence of an unforeseen chain of events in which one small act drastically impacts a large system, Wang *et al.* proposed a model for cascading failures induced by the spread of defection in networked evolutionary games. In our experiments, we implemented the evolutionary version of PDG on a structured population embedded in an undirected network. Each round consists of three phases: game playing, agent removal, and strategy update, and rounds are repeated until the population reaches a steady state (no change in agents’ strategies or survival), where we observe the survival rate ρ . In the following sections, we shall refer to vertices of the network as “nodes” when emphasizing their structural properties, and as “agents” when emphasizing cooperative or evolutionary dynamics.

2.1 Game Playing and Payoffs

In evolutionary PDG, the agents are pure strategists, i.e. each agent is either a cooperator (C) or a defector (D). All possible pairs of strategies yield four possible payoffs: R for mutual cooperation, T for unilateral defection (betrayal), S for unilateral cooperation, and P for mutual defection. PDG is characterized by the payoff structure $T > R > P \geq S$. With this payoff structure, defection is the optimal strategy for any single agent maximizing its own payoff, while cooperation is the optimal strategy for the agents population as a whole, resulting in a conflict in interests between the individual and the group. In our experiments, we set $R = 1$, $P = 0$, $S = 0$ as constants and vary $T = b$ as a parameter.

		Player 2	
		C	D
		Player 1	
Player 1	C	(1, 1)	(b, 0)
	D	(0, b)	(0, 0)

The population structure represented by an undirected network where each node represents an agent and its links determining which agents it interacts with (plays the PD game). We initialize the network with the largest hub as the sole defector, and all other agents as cooperators. Each round during the game playing phase, all linked agents play a PDG with each other. At the end of the game playing phase, we calculate each agent’s cumulative payoff π_i .

2.2 Tolerance and Agent Removal

During the agent removal phase, we assess each agent's cumulative payoff, and remove the ones with low payoffs (payoffs $< T_i$). Each agent is assigned a tolerance value T_i directly proportional to the agent's neighborhood size (i.e. number of neighbors). We set

$$T_i := \alpha k_i^0, \quad (1)$$

where $0 \leq \alpha \leq 1$ is a tolerance parameter and k_i^0 is the agent's initial number of neighbors. An agent and its links will be removed if $\pi_i < T_i$ at the end of a round.

2.3 Strategy Update

In the Strategy Update phase, each surviving agent can update their strategy to imitate a neighbor with higher payoffs. It has been found that the update rule has a non-trivial influence on the outcome of networked evolutionary games [10, 19]. Therefore, for the sake of consistency, we use the Fermi rule adopted by [17] and other literature for being analytically traceable [2, 13, 14]. At the end of each round, each survival agent i randomly chooses a neighbor j and adopts j 's strategy with probability according to Eq. 2

$$\pi_{i \rightarrow j} = \frac{1}{1 + e^{-(\pi_j - \pi_i)/\mathcal{K}}} \quad (2)$$

where \mathcal{K} is a noise parameter set constant at 0.1 as in [17]. It is worth noting that in highly connected networks with heterogeneous degrees, more games are played per round. Thus, the agents naturally receive higher cumulative payoffs. Therefore, like in real social networks, hub agents are more likely to be imitated regardless of the fitness of their strategy [15].

2.4 Results from Existing Literature

Wang *et al.* thoroughly discusses cascading failures induced by defectors in PDG played on regular lattices, and typical complex networks including scale-free and small-world networks [17]. They conclude that (1) defection strategies for temporal high reward may result in cascading failures or the collapse of the entire system and (2) the only strategy capable of surviving such failures is cooperation. Generally speaking, survival rate decreases with a higher tolerance parameter (α), while the number of stable agent clusters decreases with higher payoff to unilateral defection (b).

3 Network Topology

The two most commonly used complex network models for networked games (e.g. [9, 17]) are the Watts-Strogatz small-world (SW) model [7, 18]) and the Barabási-Albert (BA) scale-free (SF) model [1]. While these models can produce useful

insights when examining the effects of SW and SF properties individually, a unifying framework would be more valuable for our purpose of studying survival and failure in networked evolutionary games.

3.1 The Klemm-Eguíluz Growing Network Model

Due to the limitations of the SW model and the SF model alone, we choose the Klemm-Eguíluz (KE) model [3] as the standard complex model for our network topologies. The main advantage of the KE model is that it unifies important aspects of the SW and SF models: it is highly clustered, average path length between nodes can be small, and the degree distribution follows the power law. In addition, the KE model allows for safe side-by-side comparison with the high-clustering model [4] and the Barabási-Albert model.

A short description of the KE algorithm is as follows: 1. initialize a complete network of size m and activate all m nodes 2. add a new active node i to the network, and attach a link between i and each other m active node $j_1 \dots j_m$ 3. for each new link attached to j_k , rewire it to a random node with probability μ 4. deactivate a random active node according to inverse preferential selection 5. repeat steps 2–4 until the network size reaches N .

3.2 Network Parameters

Connectivity Parameter (m). In the KE algorithm, the parameter $m \in \mathbb{Z}^+ \setminus \{1\}$ determines the number of connections created by each new node (as in the original BA model), therefore the overall density of the network. Therefore, the average node degree is given by $\langle k \rangle = [\frac{1}{2}m(m - 1) + m(N - m)]/N$, with $\langle k \rangle \approx 2m$ for sufficiently large networks.

Mixing Parameter (μ) and Network Topologies. The second parameter $\mu \in [0, 1]$, often referred to as the mixing parameter, determines the probability for rewiring new connections. Notice that $\mu = 0$ results in the highly clustered (HC) topology [4], since a new node can only connect to all current active nodes, forming a long chain of highly clustered neighborhoods. Due to the absence of shortcuts, the HC topology exhibits the large-world behavior. On the other hand, $\mu = 1$ gives the BA model, since the active status is virtually ignored and growth of connections is purely based on preferential attachment. Finally, cross-over models can be obtained by setting $0 < \mu \ll 1$, producing both SW and SF properties.

In our experiments we compare four network topologies, controlled by parameter μ : the HC topology ($\mu = 0$), $\mu = 0.1$, $\mu = 0.5$, and the BA topology ($\mu = 1$).

4 Experiments

4.1 Brief Examples of Simulations

We call the whole process of agents beginning to interact until the topology reaches a steady state one **simulation**. A simulation begins with one defector

and all other agents as cooperators. For the parameters used in our experiments, we observed that simulations end up in one of two situations: all agents have died due to cascading failures or the sole survivors are cooperators that become resilient to defection.

4.2 Survival Rate w.r.t. α

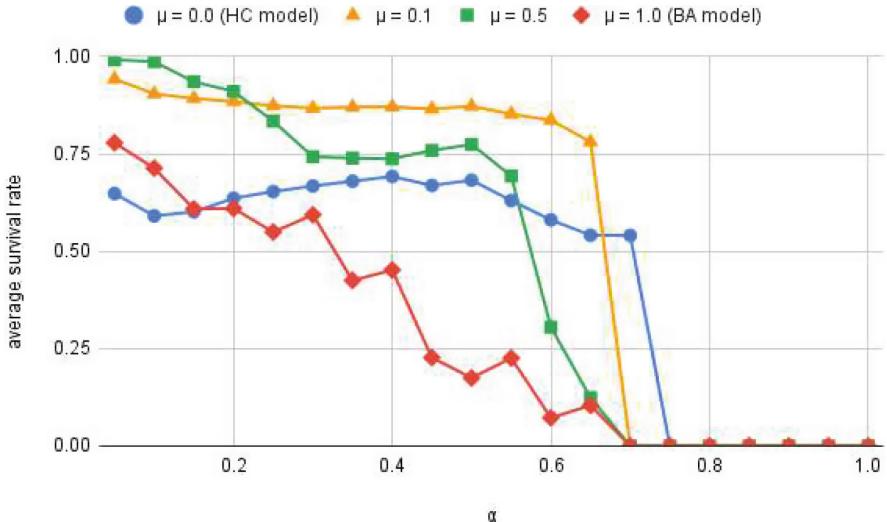


Fig. 1. Survival rate w.r.t tolerance α on four KE topologies with $N = 1000$ agents and $m = 8$. Each data point is averaged across 20 independent simulations for each of the four network topologies. Evolutionary parameter $b = 1.1$. Before passing α 's critical point, lower μ implies a more stable average survival rate.

Unsurprisingly, our simulation results in Fig. 1 show that with the exception of the HC topology, the survival rate for each topology decreases as α marginally increases by 0.05. Comparing $\mu = 0.0$ (HC topology) against $\mu = 1.0$ (BA model), we see that the survival rate decreases at a faster rate with respect to α . Notice that there is a certain tolerance value α for each topology beyond which the survival rate plummets to 0. We ascertain from this observation that the extinction boundary for each topology with the evolutionary parameters $m = 8$, $b = 1.1$ exists in terms of α between the points at which the survival rate drops to 0. For instance, the extinction boundary for the HC topology (denoted as the blue line in Fig. 1 as $\mu = 0.0$) exists in terms of α such that $0.7 < \alpha \leq 0.75$ with parameters $m = 8$, $b = 1.1$.

4.3 Survival Rate w.r.t. α and b

Figure 2 illustrates the relationship between evolutionary parameters α, b and survival rate on a cross-over topology. As expected, complete failure occurs regularly when agents have weak tolerance or the payoff to unilateral defectors is high. Again, for partial failure, cooperators are the sole survivors throughout the parameter space. Interestingly, we observe that the boundary for complete failure hardly depends on b if it is sufficiently small. In that case, the extinction boundary α^* only depends on the structure of the network. Therefore, we continue to explore the effect of network parameters on the networks' robustness against hub defection.

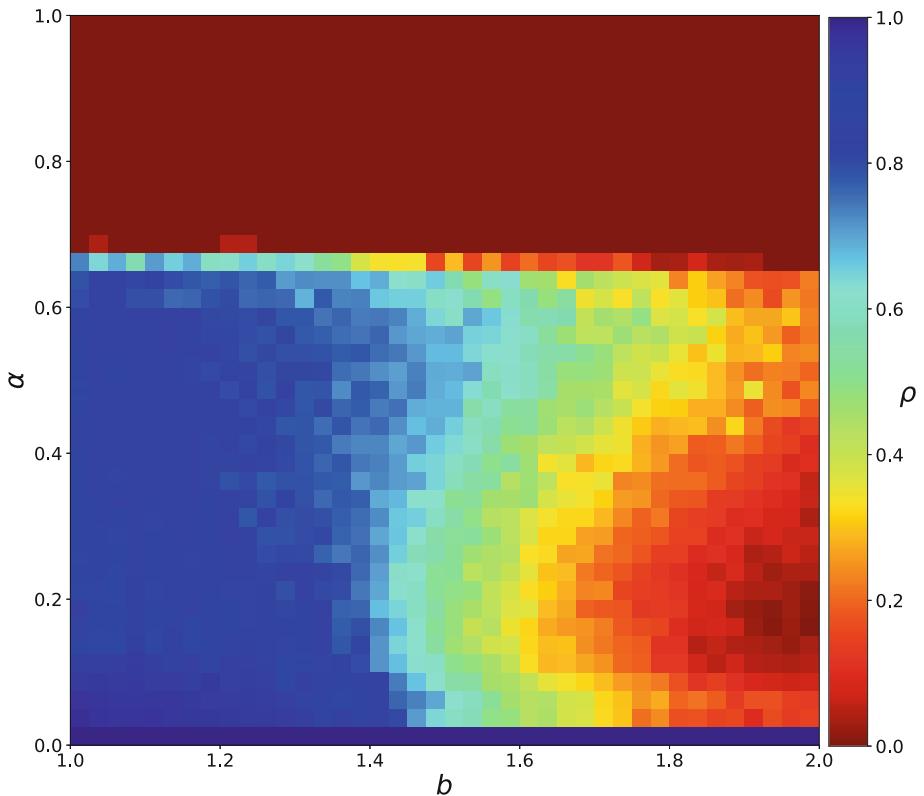


Fig. 2. Survival rate w.r.t. α and b on a KE network with $N = 1000$, $m = 10$, $\mu = 0.1$. Each colored pixel is an average of 20 independent simulations.

4.4 Survival Rate w.r.t. m

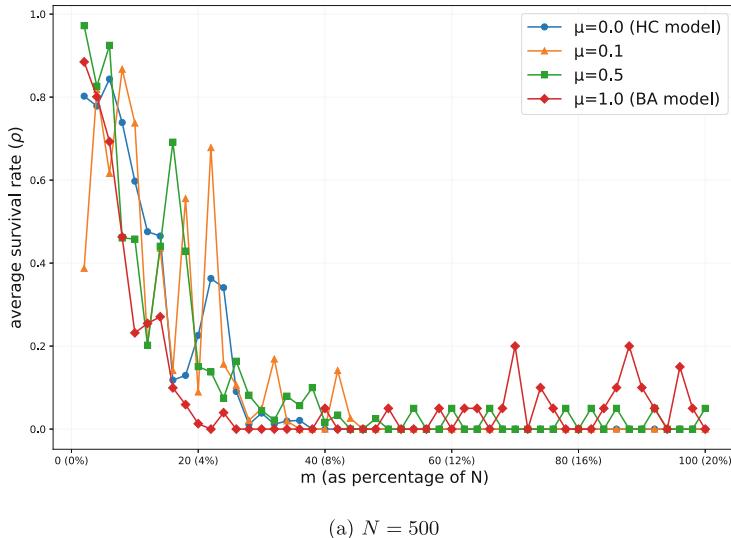
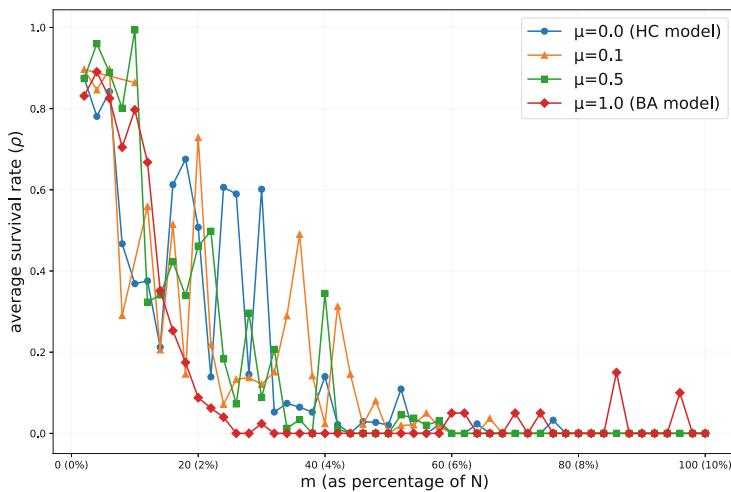
(a) $N = 500$ (b) $N = 1000$

Fig. 3. Survival rate w.r.t. m on four KE topologies with $N = 500$ and $N = 1000$. Each data point is averaged across 20 independent simulations for each network realization. At the x-axis we depict the actual m value along with its percentage of N to facilitate a comparison between $N = 500$ and $N = 1000$. Evolutionary parameters: $\alpha = 0.25$, $b = 1.1$.

Our simulation results in Fig. 3 show that for all 4 topologies and two network sizes ($N = 500$ and $N = 1000$), networks with high m values are more vulnerable to the spread of defection and prone to complete failure. Since the parameter m is closely related to the average degree $\langle k \rangle$, this result is parallel to established findings that high average degree inhibits cooperation for a wide range of networks and game dynamics [6, 11, 16, 17]. The qualitative explanation for this phenomenon is the emergence of mean-field type behaviors at high values of m . As m increases, the network becomes more similar to a completely connected network (which is equivalent to a unstructured population). In that case, defection spreads more easily from the initial defector to a major proportion of the population as in an unstructured population, eventually causing mass failure. Therefore, high connectivity leads to lower survival rate. Note that the increase in clustering due to high connectivity is not enough to counter this effect.

One caveat is that at lower m values, the effect of connectivity can be offset by variations in specific realization of the network, especially for HC and the cross-over topologies. For the same reason, BA networks can sometimes prevent complete failure even at high m values (in which case the survival rate is surprisingly high).

4.5 Extinction Boundary w.r.t. m

Based on our results in the previous section and inspired by [17], we hypothesize that given a network size and topology, the extinction boundary α^* depends only on the network connectivity parameter m , at least for small values of b . Using $b = 1.1$ on networks with $N = 1000$ and varying m , we were able to find the extinction boundary α^* numerically using a bisection algorithm. Results in Fig. 4 show that the rate with which the extinction boundary decreases varies by topology (most rapidly in the BA topology while most slowly in the HC topology). Meanwhile, the boundary values for the cross-over topologies appear lower than the HC topology while higher than the BA topology.

This result, combined with Fig. 1, leads to an interesting conclusion: populations represented by the cross-over topologies, although achieving higher survival rate when agents' are highly tolerant to low payoffs, have a lower extinction boundary compared to those represented by the HC topology.

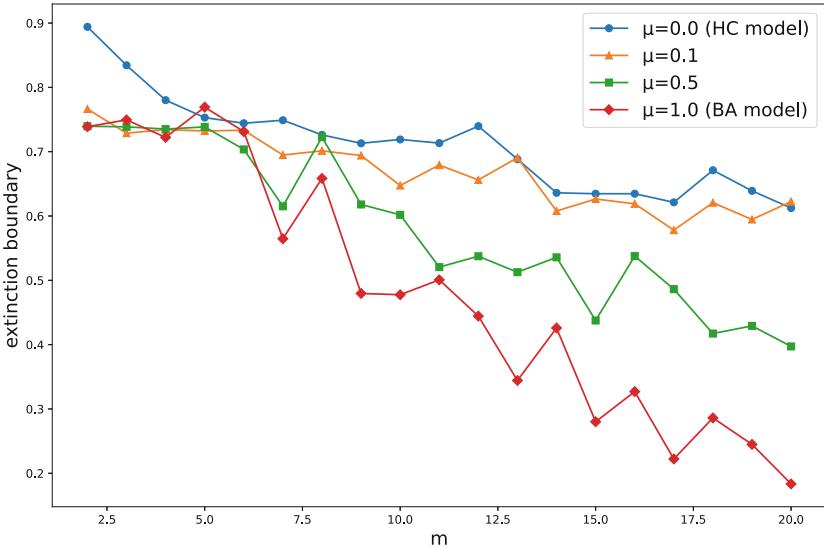


Fig. 4. Extinction boundary w.r.t. m . Controlled parameters: $N = 1000, b = 1.1$

5 Final Remarks

We use numerical simulations to investigate cascading failures induced by defection in the networked Prisoner’s Dilemma game. Our experimental results show that, similar to [17], defection of a single agent can lead to large-scale cascading failure of the network, and the sole survivors, if any, would be cooperators. Our work contributes to the understanding and generalization of evolutionary PDG on complex networks, mainly by incorporating a unified scale-free model (the Klemm-Eguímez model) with tunable parameters capable of generating structured and unstructured scale-free networks.

Comparing the results in Figs. 1 and 4, we consider the parameters used to run experiments for the former ($m = 8, b = 1.1$). We theorized above that for the HC topology ($\mu = 0.0$), there exists an extinction boundary in terms of α such that $0.7 < \alpha \leq 0.75$ given the parameters. Observing the extinction boundary value results in Fig. 4, our theory seems to hold. At $m = 8$ for the same HC topology, we see that this corresponds to an exact extinction boundary that lies between 0.7 and 0.75. Furthermore, we may draw the same conclusions (using the same parameters enumerated above) under the BA ($\mu = 1.0$).

Interesting results arise about the robustness of the network topologies using two different criteria. Keeping everything else equal, the BA topology ($\mu = 1$) has the lowest survival rate when survival rate is non zero, and experiences the fastest decrease in extinction boundary as network connectivity increases. On the other hand, robustness of the HC topology ($\mu = 0$) is largely insensitive to parameters m and α . The cross-over topologies $\mu = 0.1, 0.5$ bear resemblance to both the HC topology and the BA topology. Their survival rate sensitivity to

higher tolerance appears to be in between the HC and BA topologies, but both having higher survival rates at lower α values. In terms of extinction boundary, the boundary value for cross-over topologies exist in between the HC and BA topologies.

In summary, here we investigated how different Klemm-Eguíluz networks responded to the PDG starting from a single defector placed in a hub and how the defective strategy spread through the network, often leading to a cascading failure effect. For future work, we intend to expand the number of nodes and combinations of μ and m values. By increasing the number of independent simulations, we would be able to conduct more rigorous analysis on the distribution of survival outcomes, thus attain higher statistical significance. The model dynamics can be extended to incorporate game dynamics other than the PDG or strategies other than pure cooperation and defection. We also aim to examine other choices for placing the single defector at the start of a simulation. Regarding fields of application, we may examine these network dynamics using reinforcement learning techniques for agent-agent interactions. This work can also be relevant to the study of pandemics, financial crises, and evolutionary biology.

Acknowledgements. The authors would like to thank Grinnell College's Mentored Advanced Projects ([MAP program](#)).

References

1. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* **286**(5439), 509–512 (1999). <https://doi.org/10.1126/science.286.5439.509>
2. Hauert, C., Doebeli, M.: Spatial structure often inhibits the evolution of cooperation in the snowdrift game. *Nature* **428**(6983), 643–646 (2004)
3. Klemm, K., Eguíluz, V.M.: Growing scale-free networks with small-world behavior. *Phys. Rev. E* **65**(5), 057,102 (2002). <https://doi.org/10.1103/PhysRevE.65.057102>
4. Klemm, K., Eguíluz, V.M.: Highly clustered scale-free networks. *Phys. Rev. E* **65**(3), 036,123 (2002)
5. Kuhn, S.: Prisoner's Dilemma. In: Zalta, E.N. (ed.) *The Stanford Encyclopedia of Philosophy*, Winter, 2019th edn. Stanford University, Metaphysics Research Lab (2019)
6. Li, M., O'Riordan, C.: The effect of clustering coefficient and node degree on the robustness of cooperation. In: 2013 IEEE Congress on Evolutionary Computation, pp. 2833–2839. IEEE (2013)
7. Newman, M.E.J., Watts, D.J.: Scaling and percolation in the small-world network model. *Phys. Rev. E* **60**(6), 7332–7342 (1999). <https://doi.org/10.1103/PhysRevE.60.7332>
8. Nowak, M.A., May, R.M.: Evolutionary games and spatial chaos. *Nature* **359**(6398), 826–829 (1992)
9. Ohtsuki, H., Hauert, C., Lieberman, E., Nowak, M.A.: A simple rule for the evolution of cooperation on graphs and social networks. *Nature* **441**(7092), 502–505 (2006)
10. Roca, C.P., Cuesta, J.A., Sánchez, A.: Evolutionary game theory: temporal and spatial effects beyond replicator dynamics. *Phys. Rev. E* **6**(4), 208–249 (2009)

11. Santos, F.C., Pacheco, J.M.: Scale-free networks provide a unifying framework for the emergence of cooperation. *Phys. Rev. Lett.* **95**(9), 098,104 (2005)
12. Santos, F.C., Rodrigues, J., Pacheco, J.M.: Graph topology plays a determinant role in the evolution of cooperation. *Proc. Royal Soc. B: Biol. Sci.* **273**(1582), 51–55 (2006)
13. Szabó, G., Fath, G.: Evolutionary games on graphs. *Phys. Rep.* **446**(4–6), 97–216 (2007)
14. Szabó, G., Hauert, C.: Phase transitions and volunteering in spatial public goods games. *Phys. Rev. Lett.* **89**(11), 118,101 (2002)
15. Szolnoki, A., Perc, M., Danku, Z.: Towards effective payoffs in the Prisoner’s Dilemma game on scale-free networks. *Physica A: Stat. Mech. Appl.* **387**(8–9), 2075–2082 (2008)
16. Tang, C.L., Wang, W.X., Wu, X., Wang, B.H.: Effects of average degree on cooperation in networked evolutionary game. *Eur. Phys. J. B-Condens. Matter Complex Syst.* **53**(3), 411–415 (2006)
17. Wang, W.X., Lai, Y.C., Armbruster, D.: Cascading failures and the emergence of cooperation in evolutionary-game based models of social and economical networks. *Chaos* **21**(3), 033,112 (2011). <https://doi.org/10.1063/1.3621719>
18. Watts, D.J., Strogatz, S.H.: Collective dynamics of ‘small-world’ networks. *Nature* **393**(6684), 440–442 (1998)
19. Wu, Z.X., Guan, J.Y., Xu, X.J., Wang, Y.H.: Evolutionary Prisoner’s Dilemma game on Barabási-Albert scale-free networks. *Physica A: Stat. Mech. Appl.* **379**(2), 672–680 (2007)



Effect of Peer Influence and Looting Concerns on Evacuation Behavior During Natural Disasters

Matthew Hancock¹, Nafisa Halim², Chris J. Kuhlman¹⁽⁾, Achla Marathe¹, Pallab Mozumder³, S. S. Ravi¹, and Anil Vullikanti¹

¹ University of Virginia, Charlottesville, VA 22904, USA

{mgh3x, cjk8gx, achla, ssravi, vsakumar}@virginia.edu

² Boston University, Boston, MA 02218, USA

nhalim@bu.edu

³ Florida International University, Miami, FL 33199, USA

mozumder@fiu.edu

Abstract. We study evacuation dynamics in a major urban region (Miami, FL) using a combination of a realistic population and social contact network, and an agent-based model of evacuation behavior that takes into account peer influence and concerns of looting. These factors have been shown to be important in prior work, and have been modeled as a threshold-based network dynamical systems model (*2mode-threshold*), which involves two threshold parameters—for a family’s decision to evacuate and to remain in place for looting and crime concerns—based on the fraction of neighbors who have evacuated. The dynamics of such models are not well understood, and we observe that the threshold parameters have a significant impact on the evacuation dynamics. We also observe counter-intuitive effects of increasing the evacuation threshold on the evacuated fraction in some regimes of the model parameter space, which suggests that the details of realistic networks matter in designing policies.

Keywords: Network science · Graph dynamical systems · Agent-based models · Natural disasters · Evacuation

1 Introduction

Background and Motivation. The 2020 Atlantic hurricane season produced 30 named storms, of which 14 developed into hurricanes, and 7 intensified into major hurricanes. It was the most active season on record and the fifth consecutive above-average season since 2016. Total estimated costs to the U.S. from hurricanes and tropical storms in 2020 was \$95 billion, which was the 4th largest inflation-adjusted annual cost and more than twice the 41 year average of \$45.7 billion since 1980 [16]. See [13] for more details about hurricane season 2020 and others. Over the years, hurricanes are becoming more frequent and more intense, and inflict major physical and economic damage annually [15].

Timely evacuation during a hurricane can lead to greater safety and may even save lives. Not heeding evacuation orders can not only put individuals' lives at risk but also the lives of first responders. Power outage, flooding, issues with water supply, access to internet, food, etc. can make it hard to survive without help from emergency workers. However, there are many factors that go into a family's decision of whether or not to evacuate in the face of an oncoming hurricane. These can be broken into categories such as storm characteristics, family demographics, geography, and risk perceptions. Two factors that are of particular importance are: (i) evacuation behavior of peers, which can influence others to evacuate and (ii) concerns about looting and crime, if too many people evacuate from the neighborhood. Looting has a countering effect to peer influence on evacuation: concerns over looting in a depopulated area may inhibit families from leaving who would otherwise evacuate [9].

Our Contributions. We study the impact of peer influence and looting on evacuation during a natural disaster, using a detailed agent-based simulation (ABS) of a network dynamical system model. While the underlying network is realistic, the dynamical system model is stylized, and our goal is to understand its phase space properties. In another paper submitted to this conference, we study how survey data can be combined with agent based models; the analysis in this paper can help in validation of such methods. This paper takes a more mechanistic approach to evacuation modeling; the other paper takes a more data-driven approach. The former approach has the benefit that mechanistic models are typically more transferable to other situations (e.g., different hurricanes in different cities); the latter has the benefit that the model is guided by real data. Our ultimate goal is to combine these two approaches, but each is significant in its own right. Our specific contributions in this work are summarized below.

1. Study of evacuation behavior in a large urban region using detailed agent-based models. We develop a detailed agent-based model (ABM) for evacuation in Miami, FL. This combines a high resolution population and social contact network of Miami, with an ABM of evacuation. The population model integrates diverse kinds of commercial and open source datasets, including U. S. Census, American Community Survey (ACS), Public Use Microdata Sample (PUMS), National Household Travel Survey (NHTS), transportation network data, and land use data (see a summary of this process in [2,5]). Our agent-based evacuation behavior model represents peer influence and concerns of looting, as in [9], but considers more realistic parameter ranges.

2. Simulation based analysis of evacuation behavior. First, we provide some terminology; these concepts are detailed in Sect. 3. The evacuation threshold η_{min} (respectively, the looting threshold ($\eta_{min} + \eta_c$)) is the minimum (respectively, maximum) fraction of neighbors of a non-evacuating family (node) v_i in the social network G that must be in the evacuating stating for v_i to have a non-zero probability of evacuating. Hence, η_c is the regime over the range $[0, 1]$ in fraction of neighbors evacuating where a family has a non-zero probability of evacuation.

We present several new findings. First, in past work [9], the evacuation threshold η_{min} was taken as a fixed value, slightly greater than zero (value 0.0001), which fostered spread of contagion. In this work we systematically study the effect of η_{min} , and demonstrate a somewhat counterintuitive result: for a fixed η_c , the fraction of evacuating families can increase as η_{min} increases, if the non-zero probability of evacuation $p_{e,max}$ is sufficiently large. Other findings include, a precipitous drop in the fraction of evacuating families for increases in η_{min} over a narrow range, suggestive of a phase transformation. This phenomenon is robust across numbers of seed nodes and probabilities of evacuation $p_{e,max}$. Finally, the effect of η_c can be large or small; it is largest for lesser $p_{e,max}$ and lesser η_{min} . But noteworthy is that the effect of η_c captures the primary difference between the looting model and a standard threshold model (see Sect. 3 for details), because as $\eta_c \rightarrow (1 - \eta_{min})$, the looting model transforms to the standard relative threshold model. We find that for conditions studied in this work, evacuation rates saturate and the peak fraction of evacuations on any given day saturates at much lesser values of η_c , i.e., saturation of the looting model behavior to the behavior of the standard relative threshold model occurs for $\eta_c \ll (1 - \eta_{min})$.

3. Spatial extent of evacuating families. The population model provides detailed spatio-temporal information for all individuals. We zoom in on the highest household density region of Miami to evaluate spatial differences in evacuation rates. We grid this region in to roughly 20,000 rectangular cells and compute the average evacuation rate of all families within each cell, and plot these results as heatmaps. Quite surprisingly, we find that although there are variations, the average evacuation rates are largely homogeneous in space.

Novelty of Our Work. Our work is the first to study the role of peer influence and looting behavior on a high resolution synthetic population in a major urban region. While the specific dynamical system model we use here was proposed in prior work [9], their analysis was restricted to a smaller region (Virginia Beach, VA), and used a stylized Kleinberg small-world network, instead of more realistic contact structure based on diverse transportation and land use datasets, as we do here. We find that the realistic network structure does have an impact on the observed evacuation behavior. We also find the model is quite sensitive to the parameters, which was not studied before. Our work points to the importance of representing realistic populations in such analyses. Other (urban) populations can be studied.

Related Work. We divide related work into the following two themes.

Factors Influencing Evacuation Decisions. Factors important to families in evacuation decision-making are: receiving an evacuation notice, traffic gridlock, presence of children and pets, age of decision-makers, the household's education level, property protection, household income, work duties, race, availability of resources, and having somewhere to stay [12, 14, 17]. According to [3, 6], the most important factors in choosing to evacuate are past evacuation experience, effective communication of the risks of staying, and social influences such as support networks and watching peers evacuate. The most important factors in choosing

not to evacuate are territoriality, wanting to protect from looters, underestimating the severity of the hurricane, and overestimating household safety [14]. Extensive media reporting of looting and rioting during Hurricane Katrina caused officials to strongly warn against it in subsequent natural disasters [7].

Evacuation Models and Simulations. In [18], an ABM models hurricane evacuation in the Miami-Dade area for a hypothetical category-4 hurricane. Focus is placed on shadow evacuation, or evacuation occurring outside of the mandatory evacuation zone. An ABM based on survey data models Hurricane Sandy in northern New Jersey [19]. In [12], an evacuation model was developed from survey data from persons' experiences of Hurricane Sandy. This model includes factors of peer influence, looting, and household demographics. The work closest to our is [9]; our extensions of that work are covered in the novelty section.

2 Family Social Contact Network

We briefly summarize the realistic population and social contact network model for Miami, FL, which we use in our paper; we refer to [2, 5] for complete details about this model. Each individual in the population is represented. Individuals are organized into households, which are geolocated. A census of these households at a block group level is statistically indistinguishable from the U.S. census. Each individual has a normative daily activity schedule, e.g., work, school, home, and shopping activities etc., with each activity being assigned a location and a start and an end time. Two people who visit the same location, with overlapping times of visit, are assumed to come into contact with each other.

In this way, an individual-based social contact network is generated: each individual is a node in the network, and an undirected edge is placed between two nodes if they visit the same location and their visits overlap in time. In our context, contact between individuals implies communication between individuals about evacuation and an opportunity to influence each other. For evacuation related decision-making purposes, we assume that only those people in the age range 18 to 70, inclusive, are relevant and hence only consider edges in the network between such persons. Younger (and older) individuals that might have additional information from sources like social media are not taken into account.

Table 1. Structural properties of a family-based social contact network of Miami, FL. Nodes are families and edges are interactions between families. Properties computed with the cyberinfrastructure net.science [1].

Network	Num. nodes	Num. edges	Avg. degree	Max. degree	Avg. clust. coeff.	Diameter
Miami, FL	1,702,038	42,789,880	50.3	760	0.045	9

Since evacuation decisions are made at the family level, we construct a family social contact network from the individual-based network. In this network, nodes are now families. Two families interact (communicate, form an undirected

edge) if there is at least one edge between one person in one family and one person in the second family in the individual-based social network, and both are between 18 and 70 years old. This network has 1.70 million nodes (families) and 42.8 million edges; see Table 1 for selected structural properties. Data used to generate the individuals, the demographics (e.g., age, gender), family compositions, and activity patterns and locations include, but not limited to, American Community Survey (ACS), Public Use Microdata Sample (PUMS), National Household Travel Survey (NHTS), HERE (here.com), National Center for Education Statistics (NCES), U. S. Census data, Dun and BradStreet, and Open Street Maps. See [2, 5] for details.

3 Models

3.1 Network Model

The **family social contact network** (FSCN) is the graph $G(V, E)$, with node set V and edge set E , where each family $v_i \in V$, $i \in \{1, 2, \dots, n\}$ is a node in the graph and $n = |V|$. An undirected edge is placed between two nodes v_i and v_j to form edge $e_{ij} = \{v_i, v_j\}$, $e_{ij} \in E$ (i.e., v_i and v_j communicate) if and only if at least one family member of v_i is co-located with at least one family member of v_j , and both family members are in the age group 18–70.

3.2 Peer Influence Contagion Models

Figure 1 contains the two models used in this study to quantify a family’s decision-making process in determining whether it will evacuate in the face of an on-coming hurricane, on any particular day. The models are general so that each family v_i can have different properties such as threshold $\eta_{min,i}$ and maximum daily probability of evacuation $p_e = p_{e,max,i}$. However, for this paper, we use homogeneous properties so that $\eta_{min,i} = \eta_{min}$ and $p_{e,max,i} = p_{e,max}$, etc., for all $i \in \{1, 2, \dots, n\}$.

In both models, a node at each time t has a state in $K = \{0, 1\}$. For node v_i , state $s_i = 0$ means that v_i is not evacuating; state $s_i = 1$ means that v_i is evacuating. For each v_i , there is a function $f_i: K^{d(v_i)} \rightarrow K$ that defines the process by which a node changes state from $s_i = 0$ to $s_i = 1$. Once a node reaches state 1, it stays in state 1, which is a progressive threshold model [10]. The two models differ in the forms of f_i .

The first model is the *standard-threshold* model in Fig. 1a, and is based on the relative threshold model in [4]. Each node v_i has an *evacuation threshold* $\eta_{min} \in [0.0, 1.0]$ and denotes the minimum fraction of its neighbors that must be in state 1 (i.e., evacuating) in order for v_i to change state from $0 \rightarrow 1$. Let η_1 be the fraction of neighbors of v_i that are in state 1. At each time t , for each node v_i in state $s_i(t) = 0$, the function f_i outputs $s_i(t+1) = 1$ with probability $p_{e,max}$ if $\eta_1 \geq \eta_{min}$; else f_i outputs $s_i(t+1) = 0$.

The second model, in Fig. 1b, the *2mode-threshold* model, incorporates the effect of concern for looting, which has been identified as a factor in hurricane evacuation [8]. This model also has an evacuation threshold η_{min} . It is clear from Fig. 1 that this model has a smaller range on the x-axis over which the probability of evacuation is non-zero, compared to that of the *standard-threshold* model. Specifically, families can be concerned that when many of their neighbors have already evacuated, the neighborhood is more vulnerable to looting and crime, and hence do not evacuate. Consequently, when too many families have already evacuated, the *looting threshold*, denoted by $(\eta_{min} + \eta_c)$, defines the fraction η_1 when a family's p_e drops back to zero. Therefore, at each time t , for each node v_i in state $s_i(t) = 0$, the function f_i outputs $s_i(t+1) = 1$ with probability $p_{e,max}$ if $\eta_1 \in [\eta_{min}, \eta_{min} + \eta_c]$; else f_i outputs $s_i(t+1) = 0$. This is referred to as the *2mode-threshold* model, reflecting the influence of two modes: peer influence to evacuate and looting concern to not evacuate.

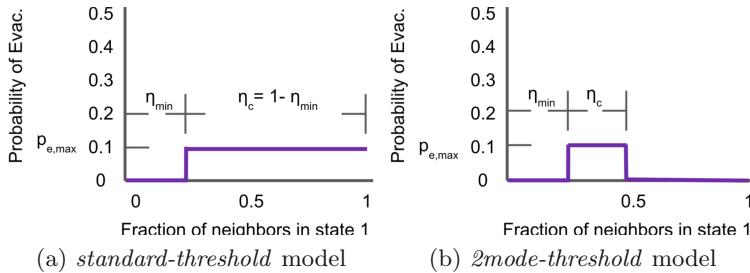


Fig. 1. Agent threshold models. The x-axis (abscissa) is η_1 , the fraction of a node's (agent's) neighbors that are in state 1. The y-axis (ordinate) is p_e , the *daily* probability that a family evacuates. (a) Classic relative threshold model where the relative threshold η_{min} is the value of η_1 where the daily probability of evacuation becomes non-zero. This is called herein the *standard-threshold* model. (b) New relative threshold model where there is a concern over looting, where the daily probability of evacuation $p_{e,max}$ returns to zero at $\eta_1 = \eta_{min} + \eta_c$. This is called herein the *2mode-threshold* model.

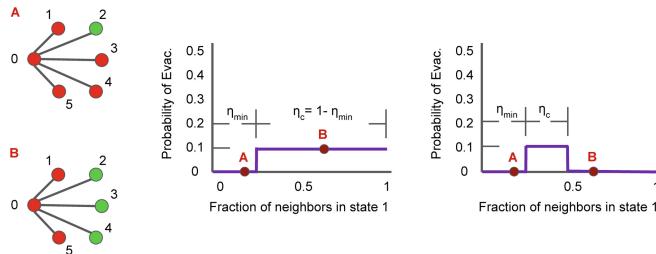


Fig. 2. (Left) Two configurations showing the five neighbors of node 0 that are in state 0 in red and in state 1 in green, labelled A and B. (Center) For the *standard-threshold* model, as η_1 increases, in going from A to B on left, p_e increases from 0 to $p_{e,max}$. (Right) For the *2mode-threshold* model, as η_1 increases, in going from A to B on left, p_e remains at 0; the block wave where $p_e = p_{e,max} > 0$ is missed.

Examples. Figure 2 provides an example of the behavior of the two models for a given ego node $v_0 = 0$ that is in state $s_0(t) = 0$. On the left are two subgraphs labeled A and B. Using the *standard-threshold* model (center graphic), node 0 in subgraph A has $\eta_1 = 1/5$ which is less than $\eta_{min} = 0.25$, and hence $p_e = 0$. That is, $s_0(t + 1) = 0$. In subgraph B, however, node 0 has $\eta_1 = 3/5$ which is greater than $\eta_{min} = 0.25$, and hence $p_e = p_{e,max} > 0$. That is, $s_0(t + 1) = 1$ with probability $p_{e,max}$. For the *2mode-threshold* model (right graphic), node 0 in subgraph A again produces $s_0(t + 1) = 0$ for the same reason as for the *standard-threshold* model. However, for subgraph B, the result for node 0 now changes from that for the *standard-threshold* model. Node 0 has $s_0(t + 1) = 0$ because $\eta_1 > (\eta_c + \eta_{min})$. For this last scenario, the contagion has spread through the network so fast that η_1 has “hopped over” the range $[\eta_{min}, \eta_{min} + \eta_c]$ where $p_e > 0$. This effect will arise in the simulation results.

4 Simulations and Results

4.1 Simulation Description and Parameters

A **simulation instance** consists of a set of **seed nodes** that are in state 1 at time $t = 0$. Time progresses forward in integer time steps (each representing one day), and at each time, each node v_i in state $s_i(t) = 0$ has its local function f_i executed, in parallel, to determine its next state, i.e., $s_i(t + 1)$. If $s_i(t) = 1$, then $s_i(t + 1) = 1$, i.e., a node that reaches state 1 remains in that state. Simulation instances are run in the interval $t \in [0, 9]$ to produce $s_i(1)$ through $s_i(10)$ for all $v_i \in V, 1 \leq i \leq n$. A **simulation** consists of a group of simulation instances. Here, we run 100 instances, each instance having a different seed node set; all other inputs are the same across instances. We use as output from the raw simulation results the average and standard deviation of the 100 results at each t . As described in Sect. 3, each f_i in the computation of $s_i(t + 1)$ represents a family behavior such as those in Fig. 1. Simulation parameters are given in Table 2.

Table 2. Summary of the parameters and their values used in the simulations.

Parameter	Description
Network	Miami, FL
Number of seed nodes, n_s	Values are 50, 100, 200, 300, 400, and 500. Seed nodes are chosen uniformly at random
Threshold model	The <i>standard-threshold</i> (i.e., classic) threshold model of Fig. 1a and the <i>2mode-threshold</i> model of Fig. 1b, in Sect. 3
Maximum probability, $p_{e,max}$	The maximum probability of evacuation $p_{e,max}$ of Fig. 1. This is a <i>daily</i> probability of evacuation for each family. Discrete values are 0.05 to 0.30 in increments of 0.05
Threshold, η_{min}	The minimum value of η_{min} where the probability $p_{e,max}$ becomes greater than zero. Discrete values are 0.01 through 0.09, in increments of 0.01
Active threshold range, η_c	The range in relative degree over which the probabilities $p_e = p_{e,max}$ are greater than zero. Discrete values are 0.2, 0.4, 0.6, 0.8, 0.90, and 0.95. Values of $\eta_c = 1 - \eta_{min}$ represent the <i>standard-threshold</i> model, whereas lesser values represent the <i>2mode-threshold</i> model

4.2 Simulation Results

Comparison of the *2mode-threshold* and *standard-threshold* Models. Basic simulation results in this section are provided for the *2mode-threshold* model. Figure 3a shows the fraction of families *newly evacuating* on each of the ten days leading up to hurricane arrival, for numbers n_s of seed nodes up to 500 families. Inputs for the *2mode-threshold* model are $\eta_{min} = 0.05$, $\eta_c = 0.2$, and $p_{e,max} = 0.3$. The fraction of evacuating families is initially small, growing noticeably over the last four days. Error bars denoting \pm one standard deviation indicate that the scatter across the 100 simulation instances is relatively small. Figures 3b and 3c show cumulative fraction of evacuating families for the *2mode-threshold* and *standard-threshold* models, respectively. The former plot plateaus for greater numbers of seed nodes, reflecting the looting threshold effect, while the latter plot shows no such constraint.

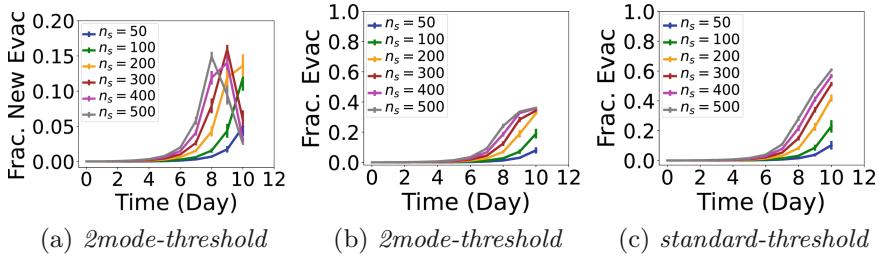


Fig. 3. Simulation results of the fraction of evacuating families in Miami, FL versus time. Fractions of (a) *newly evacuating* families and (b) *cumulative evacuating* families as a function of time for the *2mode-threshold* model. Model conditions are $\eta_c = 0.2$, $\eta_{min} = 0.05$, and $p_{e,max} = 0.30$. (c) Fraction of cumulative evacuating families for the *standard-threshold* model, with the same properties, except that $\eta_c = 1 - \eta_{min} = 0.95$. All data points on all plots display \pm one standard deviation; the variability over 100 simulation instances is not large.

Effect of Evacuation Threshold η_{min} . Figure 4 shows the effect of η_{min} for the *2mode-threshold* model for three values of $p_{e,max}$. Figures 4b and 4c show that there is a precipitous dropoff in evacuation fraction over a narrow range in η_{min} , indicative of a phase transition. These plots demonstrate that this phenomenon is persistent across different values of $p_{e,max}$ and n_s .

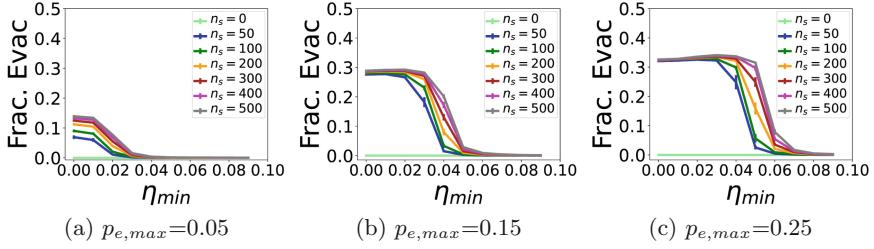


Fig. 4. Effect on cumulative evacuation fraction of η_{\min} for the *2mode-threshold* model at day $t = 10$. Here, $\eta_c = 0.2$ and $p_{e,\max}$ is (a) 0.05 (b) 0.15, and (c) 0.25. For larger $p_{e,\max}$, the evacuation fraction changes markedly over a small range in η_{\min} , suggesting a phase transformation. All data points on all plots display one standard deviation error bars; the variability over 100 simulation instances is not large.

Effect of Range η_c Over Which $p_{e,\max} > 0$. Figure 5 shows the effect of η_c on evacuation fraction (at day 10). The left-most plot is the largest fraction of newly evacuating families on any day; the remaining two plots are cumulative fractions of evacuating families. Most curves in Fig. 5a are flat for $\eta_c > 0.4$, indicating a saturation in behavior at larger η_c . This effect is also observed in Fig. 5c, for a greater value of η_{\min} . Even in the middle plot, the curves saturate at lesser η_c for $p_{e,\max} \leq 0.10$, but exhibit more changes in evacuation fraction with η_c for $p_{e,\max} > 0.1$. The point is that the two models—*2mode-threshold* and *standard-threshold* models—have as their primary difference the values of η_c . These results demonstrate that behavior of the *2mode-threshold* model can be the same as that of the *standard-threshold* model for values of η_c that are far less than $\eta_c = 1 - \eta_{\min}$, as is the case for the *standard-threshold* model. This effect is not observed over all conditions, but nonetheless over a significant range of conditions.

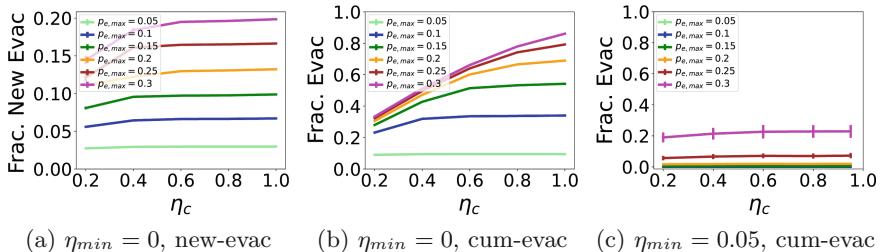


Fig. 5. Effect of η_c on the *2mode-threshold* model predictions of evacuation rates in Miami, FL. Each data point is over the 10 days leading up to hurricane landfall. Here, $n_s = 100$ families and $p_{e,\max}$ is given in legends. (a) Maximum fractions of new evacuations on any day for $\eta_{\min} = 0$. (b) Cumulative evacuation curves for $\eta_{\min} = 0$. (c) Cumulative evacuation curves for $\eta_{\min} = 0.05$. All data points on all plots display \pm one standard deviation.

Combined Effect of Evacuation Threshold η_{min} and Maximum Probability $p_{e,max}$. Figure 6 shows the combined effect of variations in η_{min} and $p_{e,max}$ on evacuation fraction. The effect is somewhat surprising in that it can be considered counterintuitive. Figures 6a and 6b depict, respectively, the largest fraction of newly evacuating nodes, on any day $t \in [1, 10]$, and the cumulative evacuation fraction. Both plots show curves that are *not* non-increasing, which is at first counterintuitive. The curve $p_{e,max} = 0.3$ in Fig. 6b, for example, shows a discernible *increasing* trend in evacuation fraction as η_{min} increases in the range $[0.01, 0.05]$. One would suspect that these curves would be non-increasing as in Figs. 4a and 4b because increasing η_{min} means that a node v_i in state 0 with small fractions of neighbors in state 1 would not produce $p_e = p_{e,max} > 0$ (i.e., a non-zero probability of evacuation).

The reason for this behavior is as follows. Data (not provided here for space reasons) demonstrate that the speed of contagion spread increases with $p_{e,max}$. Thus, for a node v_i in state 0, increasing $p_{e,max}$ causes more neighbors of v_i to change to state 1 earlier, in a sense “flooding” the neighborhood of v_i . This means that it is advantageous to increase η_{min} so that the block wave in Fig. 1b gets moved to the right so that $p = p_{e,max} > 0$ is operative for greater η_1 . If the block wave does not move further to the right, then the behavior at the right in Fig. 2, point labeled B, will be operative and the contagion will not propagate. This reasoning suggests that the effect will become more pronounced with increasing $p_{e,max}$, which is what Fig. 6b shows.

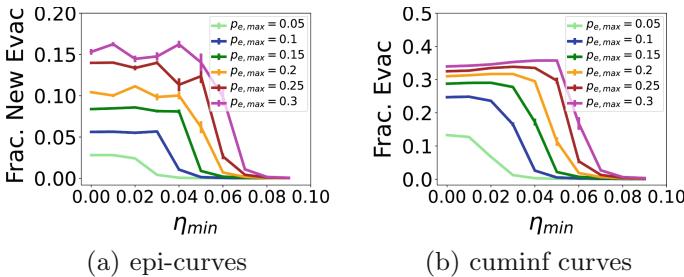


Fig. 6. Simulation results for Miami, FL. The plots are for the *2mode-threshold* model where $\eta_c = 0.2$, and η_{min} and $p_{e,max}$ vary per the plots. The number of seed nodes is 400 families. The y-axes are: (a) largest fraction of families evacuating on any day $t \in [1, 10]$, and (b) cumulative fraction of families evacuating. These plots show counterintuitive results as described in the text.

Spatial Aspects of Evacuation. Figure 7 contains heat maps of evacuation probabilities for the high population density region of Miami, FL. The maps are generated as follows. Each family dwelling is geo-located. The high population density region of the city of Miami is gridded into 156 cells in the horizontal and 137 cells in the vertical directions, producing 21372 rectangular cells. For each

of the families v_i within a cell, the fraction ρ_i of the 100 simulation instances in which the family evacuates is computed. The average value for a cell j , $\rho_{cell,j}$, which is plotted, is the average value of all ρ_i whose homes are in cell j . Results at $t = 10$ days are provided for both models. Although there are differences among cells, the results indicate that at a high level, the evacuation rates across the high population density region of Miami are fairly uniform. Evacuation rates are greater for the *standard-threshold* model.

Policy Implication and Causal Explanation. First, a sensible model at an individual level (e.g., Fig. 1) may give rise to counterintuitive behavior such as that shown in Fig. 6. This is a well-known signature of “complex systems.” Second, from a practical standpoint, the takeaway is that it is important to minimize η_{min} and to maximize $\eta_{min} + \eta_c$ in order to keep $p_{e,max}$ operative over the greatest range of η_1 . This takeaway is consistent with the statement: to increase the evacuation, allay families’ concerns over looting. From a modeling perspective, this means that interventions should seek to make families’ behaviors more like that in Fig. 1a, and less like that in Fig. 1b. And while this might seem obvious from an intuitive viewpoint, the contribution here is that this analysis provides a causal explanation. Third, the results of this paper suggest the following: government-based interventions to allay people’s concerns about looting (e.g., via increased patrols)—to increase η_c —and to incentivize evacuation—to increase $p_{e,max}$. Looting and rioting experiences from Hurricane Katrina caused police to warn against human-inflicted damage in future hurricanes [7]. This work suggested that these warnings may also encourage greater evacuation.

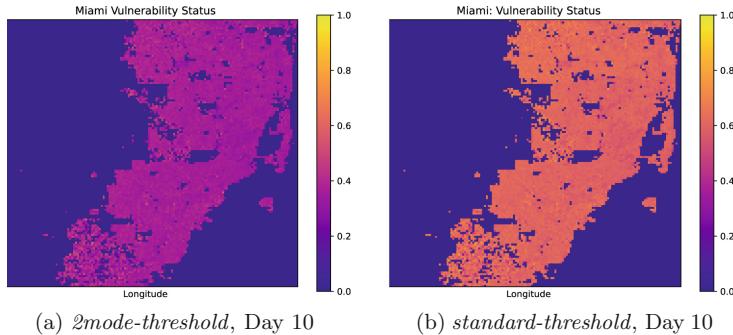


Fig. 7. Heat maps of the high population density region of Miami, representing the average evacuation rates of all families within each gridcell. There are 156×137 cells in the horizontal and vertical directions. In these plots, $\eta_{min} = 0.05$, $p_{e,max} = 0.3$, $n_s = 500$, and $\eta_c = 0.2$ and $1 - \eta_{min}$, respectively.

5 Conclusions

We present the first study of evacuation dynamics in a large urban region. Using an agent-based model, we find that peer influence and looting concerns have a significant impact on the fraction of people who evacuate. We observe the evacuation dynamics are quite sensitive to the thresholds and probability of evacuation. Our work highlights the importance of modeling detailed representations of the social network, the geospatial attributes of home locations, and realistic peer behaviors, in understanding policies and response to natural disasters. A shortcoming of our work is that we only address human contact networks; we do not include the effects of social media. We speculate that social media will not change the overall trends since the looting effect will persist. Future work includes tailoring the curves of the models in Fig. 1 for families based on household demographics. Our current plan is to make the model available in a future release of a simulation system [11] within a cyberinfrastructure [1].

Acknowledgments. We thank the anonymous reviewers for their helpful feedback. We thank our colleagues at NSSAC and Research Computing at the University of Virginia. This work has been partially supported by University of Virginia Strategic Investment Fund award number SIF160, NSF Grant OAC-1916805 (CINES), NSF CRISP 2.0 (CMMI Grant 1916670 and CMMI Grant 1832693), NSF CMMI-1745207 and NSF Award 122135.

References

1. Ahmed, N.K., Alo, R.A., et al.: net.science: a cyberinfrastructure for sustained innovation in network science and engineering. In: Proceedings of the Gateways 2020 Conference (Science Gateways Community Institute) (2020)
2. Barrett, C.L., Beckman, R.J., et al.: Generation and analysis of large synthetic social contact networks. In: Winter Simulation Conference (WSC), pp. 1003–1014 (2009)
3. Burnside, R.: Leaving the big easy: an examination of the hurricane evacuation behavior of New Orleans residents before hurricane Katrina. *J. Pub. Manage. Soc. Policy* **12**, 49–61 (2006)
4. Centola, D., Macy, M.: Complex contagions and the weakness of long ties. *Am. J. Sociol.* **113**(3), 702–734 (2007)
5. Chen, J., Hoops, S., et al.: Prioritizing allocation of Covid-19 vaccines based on social contacts increases vaccination effectiveness. medRxiv (2021). <https://www.medrxiv.org/content/early/2021/02/06/2021.02.04.21251012>
6. Cole, T.W., Fellows, K.L.: Risk communication failure: a case study of New Orleans and hurricane Katrina. *South Commun. J.* **73**(3), 211–228 (2008)
7. Faucon, C.: The suspension theory: hurricane Katrina looting, property rights, and personhood. *Louisiana Law Rev.* **70**(4), 1303–1338 (2010)
8. Halim, N., Jiang, F., Khan, M., Meng, S., Mozumder, P.: Household evacuation planning and preparation for future hurricanes: role of utility service disruptions. *Transp. Res.* **2675**, 1000–1011 (2021)
9. Halim, N., Kuhlman, C.J., Marathe, A., Mozumder, P., Vullikanti, A.: Two-mode threshold graph dynamical systems for modeling evacuation decision-making during disaster events. In: *Complex Networks*, pp. 519–531 (2019)

10. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the ACM KDD, pp. 137–146 (2003)
11. Kishore, A., Machi, L., et al.: A framework for simulating multiple contagions over multiple networks. In: Complex Networks (2021)
12. Kuhlman, C., Marathe, A., Vullikanti, A., Halim, N., Mozumder, P.: Increasing evacuation during disaster events. In: Autonomous Agents and Multiagent Systems (AAMAS), pp. 654–662 (2020)
13. Masters, J.: A look back at the horrific 2020 Atlantic hurricane season; Yale Climate Connections. <https://yaleclimateconnections.org/2020/12/a-look-back-at-the-horrific-2020-atlantic-hurricane-center/>. Accessed 13 Aug 2020
14. Riad, J.K., Norris, F.H., Ruback, R.B.: Predicting evacuation in two major disasters: risk perception, social influence, and access to resources. *J. Appl. Soc. Psychol.* **29**(5), 918–934 (1999)
15. Shultz, J.M., Galea, S.: Mitigating the mental and physical health consequences of hurricane Harvey. *JAMA* **318**(15), 1437–1438 (2017)
16. Smith, A.: 2020 U.S. billion-dollar weather and climate disasters in historical context. <https://www.climate.gov/news-features/blogs/beyond-data/2020-us-billion-dollar-weather-and-climate-disasters-historical>. Accessed 20 Aug 2021
17. Wong, S., Shaheen, S., Walker, J.: Understanding evacuee behavior: a case study of hurricane Irma. <https://escholarship.org/uc/item/9370z127>
18. Yin, W., Murray-Tuite, P., Ukkusuri, S.V., Gladwin, H.: An agent-based modeling system for travel demand simulation for hurricane evacuation. *Transp. Res. Part C Emerg. Technol.* **42**, 44–59 (2014)
19. Zhu, Y., Xie, K., Ozbay, K., Yang, H.: Hurricane evacuation modeling using behavior models and scenario-driven agent-based simulations. *Procedia Comput. Sci.* **130**, 836–843 (2018)



Eigenvalues of Random Signed Graphs with Cycles: A Graph-Centered View of the Method of Moments with Practical Applications

Pau Vilimelis Aceituno^{1,2,3(✉)}

¹ Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany

² Max Planck School of Cognition, Leipzig, Germany

³ Institute of Neuroinformatics, ETH Zürich, Zürich, Switzerland

pau@ini.uzh.ch

Abstract. We illustrate a simple connection between the cycles in a graph and eigenvalues its the adjacency matrix. Then we use this connection to derive properties of the eigenvalues of random graphs with short cyclic motifs and circulant graphs with random signs. We find that the eigenvalue distributions that emerge from those structures are surprisingly beautiful. Finally, we illustrate their practical relevance in the field Reservoir Computing.

Keywords: Random matrix theory · Cycles · Motifs · Circulant graphs · Reservoir computing

1 Introduction

Networks and graphs are a very flexible modeling tool that has been crucial in studying complex systems. Of particular importance in those studies are the spectral methods, a family of tools that use the eigenvalues of the adjacency matrix of the graph and which has been applied to fields as diverse as wireless communications, ecology, biological networks, image processing, machine learning or nuclear physics [Spi07, New18]. Since in real systems the exact graph structure is not known, researchers resort to spectral random matrix theory, the mathematical field that studies the distribution of eigenvalues from a probabilistic perspective [ER05]. However, this approach requires a deep understanding of probability theory [Tao12], and its results can be difficult to understand for non-experts. Our work is based so-called method of moments used to prove Wigner's semicircle law [KK16], and extended to other families of symmetric matrices [KK16].

In this work we give an extra twist to the method of moments by looking at the underlying graph, allowing us to go beyond the symmetric case and providing an intuitive approach to study the eigenvalues of random graphs. To show the

theoretical applications of our approach, we study two families of random networks that are constructed with cyclic structures, one showcasing the effects of short cycles and the other one the effects of long cycles. Those distributions are surprisingly beautiful, showing that even though our approach is simple, it can provide interesting results. Finally, we show that those families of matrices are also of practical importance by using them in the field of Reservoir Computing.

2 Cycles and Moments of the Eigenvalue Distribution

A directed graph $G(N, E)$ is a set of N nodes and E edges, where every edge $e = (n, m)$ represents a directed connection going from node n to node m . Each edge has an associated weight $w(e)$ which represents the strength of that connection and might be negative. The graph can be represented by its adjacency matrix \mathbf{W} , an $N \times N$ matrix where every entry M_{nm} is the weight of edge (n, m) , and non-existent edges correspond to entries with value zero.

Given all cycles of length L – the paths of length L that end where they started – we define their normalized weight

$$\rho_L = \frac{1}{N} \sum_{c \in C_L} w_c \quad (1)$$

where C_L is the set of cycles of length L , and $w_c = \prod_{e \in c} w(e)$, the multiplication of weights of the edges e in cycle c . This includes cycles where edges or nodes are visited multiple times. Since the value $\sum_{c \in C_L} w_c$ is given by the entries of the power of the graphs' adjacency matrix \mathbf{W} , we can obtain ρ_L from the adjacency matrix \mathbf{W} by

$$\rho_L = \frac{1}{N} \text{tr} [\mathbf{W}^L], \quad (2)$$

where $\text{tr} [\cdot]$ is the trace of a matrix, which equals the sum of its eigenvalues, thus

$$\rho_L = \frac{1}{N} \sum_{n=1}^N \lambda_n^L \quad (3)$$

where λ_n is the n th eigenvalue of the adjacency matrix \mathbf{W} .

Equation 3 is particularly interesting as $N \rightarrow \infty$. In this limit we can consider the eigenvalues of a matrix as random i.i.d. values sampled from a probability density function $p(\lambda)$ in the complex plane \mathbb{C} . For a random graph sampled from a probability distribution with a fixed ρ_L , this value corresponds to a moment of the eigenvalue distribution,

$$\mu_L = \int_{\mathbb{C}} p(\lambda) \lambda^L d\lambda = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N \lambda_n^L = \rho_L. \quad (4)$$

3 Random Networks with Short Cyclic Motifs

Many real-world networks have an unexpectedly high number of network motifs, small subgraphs that are overrepresented with respect to randomized networks [MSOI+02]. One of the most relevant examples is the feedback loop, a well-known structure in control theory which is equivalent to a directed cycle in a network. This section shows how feedback loops shape graph spectra by studying graphs with abundant cyclic motifs. We will consider sparse random graphs sampled from a distribution where all the connections between nodes have the same probability and with weights independently drawn from $[-w, w]$ with equal probability. In order to work on the limit $N \rightarrow \infty$, we will normalize the weights with $w = \sqrt{\frac{N}{E}}$. To account for the feedback motifs, a fraction of the edges are embedded into cycles of length τ where the multiplication of the weights of edges in each cycle has a constant sign.

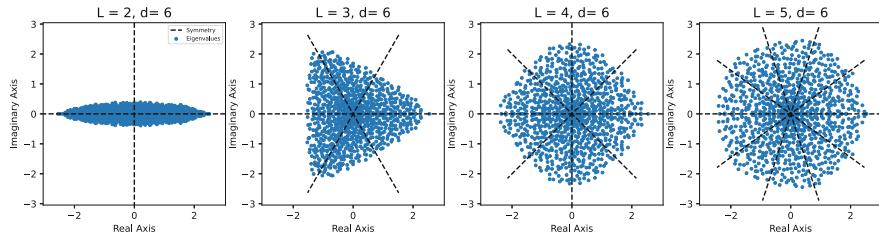


Fig. 1. Eigenvalues of networks with short cycles: Each plot shows the eigenvalues of a random graph with cycles of length τ being overrepresented. Each blue point corresponds to one eigenvalue in the complex plane and the black lines correspond to the axis of symmetry that we investigated. The graphs have 5000 nodes, average degree of 20, the connections are assigned randomly and the weights are $\left[\frac{1}{\sqrt{d}}, -\frac{1}{\sqrt{d}}\right]$ with equal probability.

Such graphs have been studied in our previous work, and the support of the eigenvalue distribution of their adjacency matrix is given by the Hypotrochoidic Law of Random Matrices [ARS19]. In this work we recover only the symmetry of that support, which is a much simpler result, but without relying on cavity methods.

When the number of nodes goes to infinity, the values of ρ_L converge to their expectations. As the weights have a symmetric distribution –meaning that a positive weight has the same probability as a negative one–, this expectation is zero except if we force it explicitly. Only the cycles of length τ have non-zero expectations, so

$$\lim_{N \rightarrow \infty} \rho_L = \begin{cases} w^L \frac{F_L}{N} & \iff L \equiv 0 \pmod{\tau} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Where C_L is the number of cycles of length L that were added as feedback loops. Since ρ_L are the moments of $p(\lambda)$, Eq. 5 implies that the moments which are not multiples of τ tend to zero. We will use this to study the symmetries in the eigenvalues, but first we need to consider the rotated density function $p(e^{\theta i}\lambda)$. The moments of this distribution can be easily computed,

$$\mu_L^\theta = \int_{\mathbb{C}} p(e^{\theta i}\lambda) \lambda^L d\lambda = \int_{\mathbb{C}} p(\lambda) \lambda^L e^{L\theta i} d\lambda = e^{L\theta i} \int_{\mathbb{C}} p(\lambda) \lambda^L d\lambda = e^{L\theta i} \mu_L. \quad (6)$$

Then the moments of $p(\lambda)$ and $p(e^{\theta i}\lambda)$ are equal when $\mu_L = 0$ or $\theta \in \{0, \frac{2\pi}{L}, \frac{4\pi}{L}, \dots\}$. By Eq. 5, $\mu_L^\theta = \mu_L \forall L \iff \theta = 0, \frac{2\pi}{\tau}, \frac{4\pi}{\tau}, \dots$. And since the eigenvalues are bounded, the equality of moments implies that

$$p(\lambda) = p(e^{\theta i}\lambda) \quad (7)$$

for $\theta \in \left\{0, \frac{2\pi}{\tau}, \dots, \frac{2(\tau-1)\pi}{\tau}\right\}$. In geometric terms, this means that $p(\lambda)$ has τ rotational symmetries in the complex plane.

Since the entries of our adjacency matrix are real, the eigenvalue distribution is also symmetric with respect to the real line. Combining this axis of symmetry with the rotational symmetries from Eq. 7, we obtain that $p(\lambda)$ has τ axes of symmetry in the complex plane, all of them passing through the origin and with at angles $0, \frac{2\pi}{\tau}, \dots, \pi - \frac{2\pi}{\tau}$.

We computed the eigenvalue distributions of large graphs with overrepresented circular motifs where degrees and weights are homogeneous in Fig. 1, which show the symmetries highlighted before.

4 Circulant Directed Networks

Having studied the spectra of random graphs with short cycles, we now investigate the effects of long cycles. For this we turn to directed circulant graphs, where every node with index $n \in \mathbb{Z}/n\mathbb{Z}$ and degree d is connected to its neighbors $[n+1, n+2, \dots, n+d]$ as shown in Fig. 2. Although the spectra of circulant graphs are well known [VM10], here we will focus on their random counterpart, which have not, to the best of our knowledge, been explored.

For simplicity, we consider random weights independently sampled from the set $\{-1, 1\}$ with equal probability. When $N \gg d$, this family of graphs contain only long cycles and is therefore well suited for our question. We will investigate the dominant eigenvalue of their adjacency matrix.

The simplest case is given when $d = 1$. Since there is only a single simple cycle of length N , $\rho_k = 0$ for all $k < N$, and $\rho_N = \prod_{n=1}^N w_{n,n+1}$. The solutions for

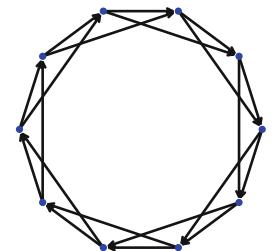


Fig. 2. A directed circulant graph with ten nodes and degree two. In the graphs studied here, the signs of the edge weights are random.

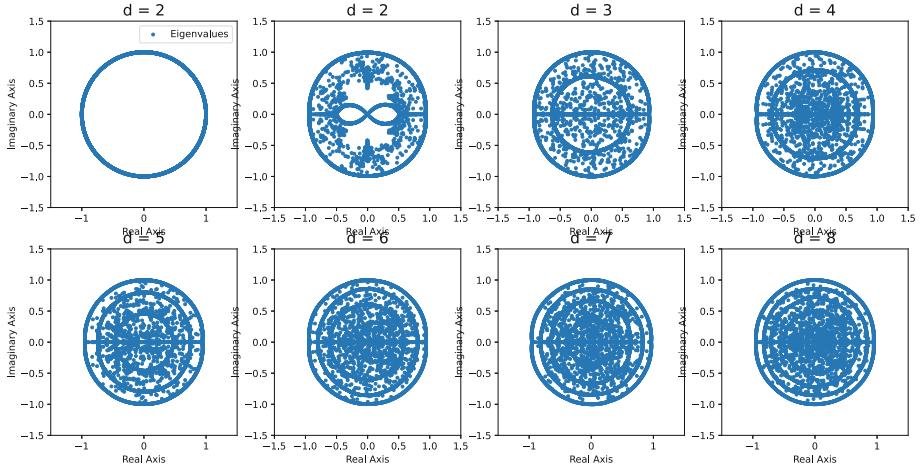


Fig. 3. Eigenvalues of circulant networks: Each plot shows the spectrum of a directed circulant graph with 5000 nodes where the weights are $[-1, 1]$ with equal probability. Each blue point corresponds to one eigenvalue in the complex plane.

those equations are the N th roots of ρ_N . For $\rho_N = 1$, $\lambda_n = e^{\frac{2\pi n}{N}i} \quad \forall n \in \mathbb{Z}/n\mathbb{Z}$ where i is the imaginary unit. If $\rho_N < 0$ the phase is shifted by $\frac{\pi}{N}$.

To expand this to $d > 1$ we take Eq. 3 in the limit where $L \rightarrow \infty$, the eigenvalues with the largest modulus dominate the rest. Then,

$$|\rho_L| = \frac{1}{N} \left| \sum_{k=1}^N \lambda_k^L \right| \sim |\lambda_{\text{Max}}|^L. \quad (8)$$

so $|\rho_L|$ grows exponentially in L at the rate λ_{Max} .

We can use Eq. 1 to write ρ_L as a binomial distribution where cycle weights w_c are independently sampled from $\{-1, 1\}$ with equal probability. Then, ρ_L can be approximated by a normal distribution $\mathcal{N}(0, \sigma_L^2)$ with $\sigma_L^2 = \frac{1}{N}|C_L|$ where $|C_L|$ is the number of cycles of length L . To obtain this value we consider the recurrence relation between paths in the circulant graph $p_L(n, m) = \sum_{k=1}^d p_{L-1}(n, m-k)$, where $p_{L-1}(n, m)$ is the number of paths from node n to m of length L . When L becomes large and $k \ll N$, $p_L(n, n) \sim p_L(n-k, n)$, so

$$\begin{aligned} |C_L| &= p_L(n, n) = \sum_{k=1}^d p_{L-1}(n, m-k) \\ &\sim dp_{L-1}(n, n) = d|C_{L-1}|. \end{aligned} \quad (9)$$

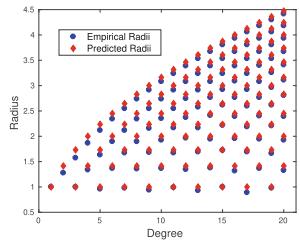


Fig. 4. Eigenvalue radii for circulant random graphs: each blue point corresponds to the radius of a circle in the distribution of eigenvalues for a circulant matrix from Fig. 3. The red rhombi is the radii given by Eq. 11, and the highest ones for every degree correspond to Eq. 10.

Given a normal distribution, the expected value of the first order-statistic has the order of magnitude of the standard deviation, so $|\rho_L| \sim \sqrt{d^L}$. Combining this with Eq. 8,

$$|\lambda_{\text{Max}}| \approx \sqrt{d}. \quad (10)$$

We computed the eigenvalues of large circulant directed networks in Fig. 3. For $d = 1$, our prediction matches the distribution. When we increase d we find a surprising result: For $d = 2$, there is an outer ring and the rest of eigenvalues concentrate in a lemniscate shape, and for $d > 2$ the eigenvalues are concentrated on the real line and on $\lceil \frac{d}{2} \rceil$ rings centered at the origin.

Equation 10 gives us the approximate position of the larger ring which agrees with the observed values (see Fig. 4). We also observe empirically that the radii of the rings are distributed following the rule

$$r_d(k) \lesssim \begin{cases} \sqrt{2k-1} & \forall d = 1 \pmod{2} \\ \sqrt{2k} & \forall d = 0 \pmod{2} \end{cases} \quad (11)$$

where $r_d(k)$ is the radius of the k th ring in a circulant graph with degree d . As we show in Fig. 4, our observation agrees with the empirical eigenvalue distribution.

5 Application: Reservoir Computing

The eigenvalues of a matrix are often useful to understand the dynamics of the system represented by the matrix. In our setting, we can use some properties of the graphs exposed here to understand existing systems or to help design new ones. Specifically, we shall study applications in Reservoir Computing, a paradigm for recurrent neural networks with applications in robotics, finance or medicine, where a large directed random network of neurons – the reservoir – is initialized used to capture features from an input time series, as explained in Appendix A.1. Given that the features to be extracted depend on the dynamics of the reservoir, and those dynamics depend on the eigenvalues of the adjacency matrix of the reservoir network, understanding the eigenvalues of a network can help us improve reservoirs.

Since the reservoir network is untrained, it is critical to make sure that it captures the right features of the input time series at initialization. In previous works [AYL20], we explored two families of features that can be used to improve reservoirs in practical tasks: memory capacity and frequency adaptation. In this section we will explore how the two families of networks presented here can be useful to design better reservoir.

5.1 Frequencies and Short Cycles

To understand the influence of circular motifs on the dynamics of a network, we must first understand that the poles of a filter correspond to the eigenvalues

of the adjacency matrix os a network (see Appendix A.2). A network is thus a very complex filter, where many poles with different phases coexist. What determines which frequencies are enhanced or damped in such a system is then the maximum eigenvalue with phase $2\pi f$. Thus, capacity of a reservoir to enhance a specific family of frequencies is given by the difference between the furthest and closest point from the origin in the contour of the hypotrochoids seen in Fig. 5.

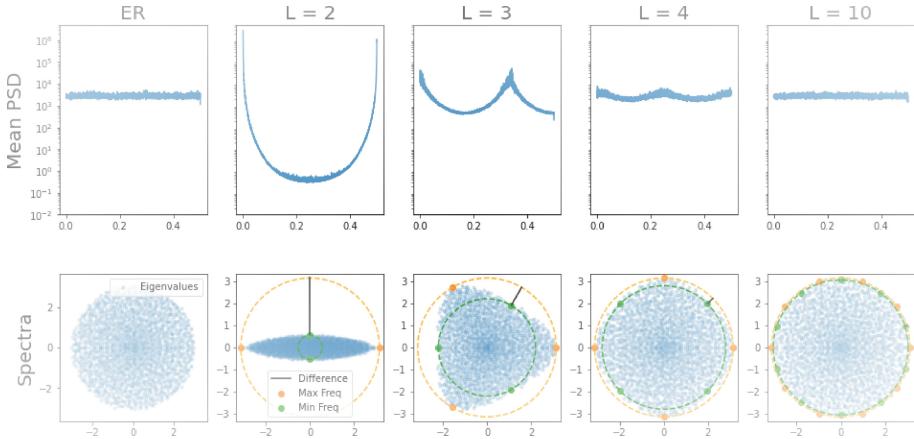


Fig. 5. Frequency responses and eigenvalue distributions for reservoir networks with cycles: In the upper row we present the Power Spectral Density (PSD) averaged across all neurons in a reservoir feed with white Gaussian noise, and on the lower row the eigenvalues of such networks with the difference between the largest phases (orange) and minimum phases (green) of the eigenvalue distribution, along with the difference between both (in black). Each column represents a different type of network, with an Erdős Rényi network (ER) on the left and random networks constructed by combining cycles of different lengths L on the rest of the columns. As L grows spectra of the networks constructed with cycles resemble those of an ER, and the frequency response (upper row) follows the same pattern.

In [AYL20], we showed that making reservoirs resonate at task-specific frequencies can help improve RC performance. This resonances can be enforced by adding cycles into the reservoir, thus the networks with short motifs are applicable in RC. While adding cycles of lengths 1, 2 and 3 can improve RC performance, we found that longer cycle lengths do not seem to have practical relevance, as networks created by composing cycles of larger lengths do not seem to enhance the frequencies as much as those based on short cycles, as seen in Fig. 5.

This problem can be understood by looking at how the eigenvalue distributions changes for cycles of high lengths, as shown in Fig. 5, where the eigenvalue distribution of the reservoir network with cycles of length L approaches that of a random network as the length L grows. Geometrically, this is because a

regular polygon – or a convex hypotrochoid – with L sides resembles a circle as L increases. As in a circular eigenvalue distribution the contour is at a fixed distance from the origin independently of the phase, the reservoirs tend to resonate more uniformly to all frequencies.

The analysis of the resonances used in [AYL20] (present in the appendix) would suggest that the decrease is due to the saturating effect of the non-linearity, while our analysis suggest that it is a network-based effect. This implies that it can and should be addressable at the structural level, which is more convenient from an engineering perspective.

5.2 Memory Maximization in Circulant Matrices

A key feature of the reservoir is its capacity to retain previous inputs, which is measured by the memory capacity

$$M = \sum_{\tau=0}^{\tau_{\max}} M_\tau \quad (12)$$

$$M_\tau = \max_{\mathbf{r}_\tau} \frac{\text{cov}^2[y_\tau(t), u(t - \tau)]}{\text{Var}[u(t)] \text{Var}[y(t)]},$$

where $y_\tau(t) = \mathbf{r}_\tau \mathbf{x}(t)$ is the best linear approximation that the reservoir can provide of the input at time $t - \tau$ at time t . Intuitively, the decay of M_τ tells us how fast does the reservoir forgets previous inputs and thus measures the memory.

The total memory capacity M is known to depend on the sum of the absolute values of the eigenvalues of the adjacency matrix of the graph [AYL20]. Intuitively, this is because the reservoir is a dynamical system that keeps traces of previous inputs in its dynamics and the eigenvalues of the adjacency matrix – which give the first order approximation of the nonlinear reservoir – tell us how close those perturbations decay: the larger the eigenvalues the more time it takes for the inputs to fade away.

Thus, a network with large memory must have large eigenvalues and consequently the classical method to increase the memory is to increase the spectral radius of

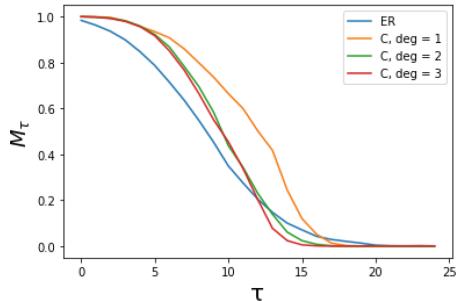


Fig. 6. Memory Decay: We plot the values of M_τ for various values of τ in the standard ER network (blue) and for circular networks with different degrees. The figure shows how the memory decays at a slower rate for circulant networks than for the ER one with a fixed amount of neurons (500) and a fixed spectral radius (1). The total memory capacity M corresponding to the area under the curves are 9.4 for ER, 11.4 for circulant with degree 1, 10.5 for degree 2, and 10.0 for degree 3

the matrix or make the matrix orthonormal [FBG16]. However, if we want to keep the reservoir in the stable regime, the spectral radius can only grow up to one. Other methods to increase the memory, such as the orthonormalization, provide memories beyond those of a random network with a spectral radius of one, but they require the network to be dense. As the number of computer operations to run the dynamics of the reservoir grows with $O(E)$ where E is the number of edges, having a sparse matrix is more desirable.

Thus, circulant matrices offer a good memory without increased computational demands: Given that most of their eigenvalues are far from the origin, the sum of eigenvalues is large for a given spectral radius. Furthermore, they only require dN edges with low values of d , hence they are very sparse. When we tested this networks empirically, we do find that their memory is indeed higher than for the standard ER networks, as shown in Fig. 6.

6 Future Works

Besides the generic contribution of understanding the structure-dynamics connection, there are a few avenues of research that span from the results presented here. On the theory side, we would like to investigate how to create networks with cyclic motifs and sharper peaks in their eigenvalue distributions – which translate into stronger resonances–, as well as the formation of rings and lemniscate shapes in circulant networks. On the practical side, we believe that our networks can help improve standard recurrent neural networks by initializing the network structure in a more adaptive way, namely by introducing the resonances or long-memories properties when the tasks to be performed are known to have such requirements. In particular, univariate time series processing problems such as detecting arrhythmias in ECG – where the frequency of the signal is well characterized can benefit from our predefined frequency adaptation. Similarly, the physical implementation of recurrent neural networks in neuromorphic hardware often suffers from fast memory decay and having a predefined structure that can maintain longer memories without requiring dense networks – which are unpractical in hardware – could be beneficial.

7 Conclusion

In this paper we have illustrated a connection between the eigenvalues of the adjacency matrices of random graphs, then presented two families of networks with surprising eigenvalue distributions that can be studied with that connection and explored the practical implications they have in Reservoir Computing.

It must be noted that there are more powerful methods to study the eigenvalues of large graphs which can give more details about the spectra of random graphs [RC09]. Also, more common machine learning methods to train recurrent neural networks such as backpropagation through time achieve better performances than reservoir computing, albeit with much higher training costs and requiring more data.

The relevance of our work relies therefore on highlighting a clear connection between a network's dynamics and its structure: cycles are easy to understand and quantify, while eigenvalues are better suited to study a systems' dynamics. By establishing a link between both, we illustrate insights into their relationship that can be exploited in other areas.

Acknowledgements. This work was supported by the Bundesministerium für Bildung und Forschung through the Max Planck School of Cognition.

A Appendix

A.1 Reservoir Computing

In the most common implementation of RC [JH04], the reservoir is described by

$$\mathbf{x}(t) = \tanh [\mathbf{W}\mathbf{x}(t-1) + \mathbf{w}_{\text{in}}u(t)] \quad (13)$$

where $\mathbf{x}(t)$ is the state of the reservoir at time t given by the output of all neurons at that time, \mathbf{W} is the adjacency matrix of the reservoir network, \mathbf{w}_{in} is a vector determining how the input is fed into the reservoir network and $u(t)$ is the input time series to be processed.

The output $y(t)$ is then given by a linear combination of the neurons' outputs with coefficients given by the readout vector \mathbf{r} ,

$$y(t) = \mathbf{r}\mathbf{x}(t) \quad (14)$$

which is usually obtained by a linear regression or similar methods.

Beyond the mathematical description, it is useful to realize that a reservoir can be seen as a set of nonlinear filters that extract features of the input signal while keeping traces of previous inputs in its dynamics and a linear readout can then be used to extract relevant information by training it with linear methods. Thus, having the right filters for a certain task – by tuning their memory or resonances, for instance – can help improve the reservoirs' performance.

A.2 Poles and Eigenvalues

Here we look at the relationship between eigenvalues and poles in a linear infinite impulse response (IIR) filter. As illustrated in Fig. 7 an IIR filter can be implemented as a cyclic network with the appropriate cycle length: the eigenvalues of the network will then correspond to the poles of the filter and the phases of the pole tell us which frequencies are enhanced while their distance to the origin tells us how strong the enhancement is.

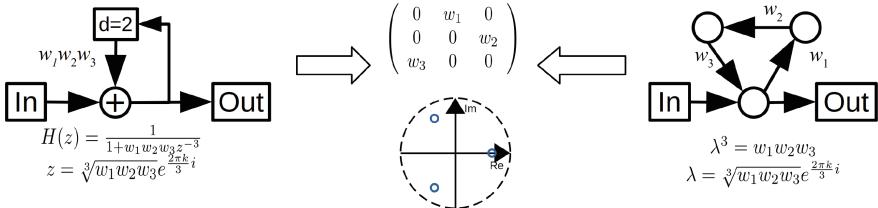


Fig. 7. Simple example of the relationship between poles and eigenvalues: An IIR filter with delay 2 and feedback weight $w_1w_2w_3$ (left) can be implemented as a cycle of length 3 with weights w_1, w_2, w_3 , and the poles of the transfer function are given by the eigenvalues of the adjacency matrix. The plot in the complex plane represents thus the poles (or eigenvalues) of this system.

References

- [ARS19] Aceituno, P.V., Rogers, T., Schomerus, H.: Universal hypotrochoidic law for random matrices with cyclic correlations. *Phys. Rev. E* **100**(1), 010302 (2019)
- [AYL20] Aceituno, P.V., Yan, G., Liu, Y.-Y.: Tailoring echo state networks for optimal learning. *iscience* **23**(9), 101440 (2020)
- [ER05] Edelman, A., Rao, N.R.: Random matrix theory. *Acta Numer* **14**, 233–297 (2005)
- [FBG16] Farkaš, I., Bosák, R., Gergel', P.: Computational analysis of memory capacity in echo state networks. *Neural Netw.* **83**, 109–120 (2016)
- [JH04] Jaeger, H., Haas, H.: Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* **304**(5667), 78–80 (2004)
- [KK16] Kirsch, W., Kriecherbauer, T.: Sixty years of moments for random matrices. arXiv preprint [arXiv:1612.06725](https://arxiv.org/abs/1612.06725) (2016)
- [MSOI+02] Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: simple building blocks of complex networks. *Science* **298**(5594), 824–827 (2002)
- [New18] Newman, M.: Networks. Oxford University Press (2018)
- [RC09] Rogers, T., Castillo, I.P.: Cavity approach to the spectral density of non-Hermitian sparse matrices. *Phys. Rev. E* **79**(1), 012101 (2009)
- [Spi07] Spielman, D.A.: Spectral graph theory and its applications. In: 48th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2007, pp. 29–38. IEEE (2007)
- [Tao12] Tao, T.: Topics in Random Matrix Theory, vol. 132. American Mathematical Society (2012)
- [VM10] Van Mieghem, P.: Graph Spectra for Complex Networks. Cambridge University Press (2010)



Bosonic Random Walk Neural Networks for Graph Learning

Shiv Shankar^(✉) and Don Towsley

College of Information and Computer Science, University of Massachusetts,
Amherst, USA
sshankar@cs.umass.edu

Abstract. The development of Graph Neural Networks (GNNs) has led to great progress in machine learning on graph-structured data. These networks operate via diffusing information across the graph nodes while capturing the structure of the graph. Recently there has also seen tremendous progress in quantum computing techniques. In this work, we explore applications of multi-particle quantum walks on diffusing information across graphs. Our model is based on learning the operators that govern the dynamics of quantum random walkers on graphs. We demonstrate the effectiveness of our method on classification and regression tasks.

Keywords: Quantum random walk · Graph neural network · Graph classification · Graph regression · Quantum learning

1 Introduction

The current era of ubiquitous connectivity has provided researchers with ever-increasing troves of data. Most of such ‘real-world’ data have an underlying graphical structure that can be utilized to build better models and derive greater insights. Such graphical structures are not limited to the web, social networks, or other network systems. Graph-structured problems are also common in many scientific fields such as immunology (Crossman 2020), chemical analysis (John et al. 2019) and bio-chemistry (Bonetta and Valentino 2019).

Current machine learning approaches for analyzing structured data can be broadly categorized into neural approaches and classical approaches. Classical approaches rely on comparing graphs by utilizing various similarity notions Kondor et al. (2009); Kondor and Borgwardt (2008). Similarity comparison between graphs can be performed directly via walks (Dobson and Doig 2003; Callut et al. 2008). Another related technique is to use graph kernels (Vishwanathan et al. 2010; Gärtner et al. 2003). There have been some recent works (Bai et al. 2017a) that try to define similarity using quantum walks.

The last decade also saw great progress in machine learning via the development of deep-learning techniques. Some of these works also focused on applying neural networks to graph-structured data (Defferrard et al. 2016; Duvenaud

et al. 2015) Duvenaud et al. (2015) present a method for differentiable finger-printing where the hashing functions are replaced by neural networks. Defferrard et al. (2016) extend the convolution operator to graphs using graph Laplacians. Kipf and Welling (2017) use the same technique for semi-supervised learning on graphs. Atwood and Towsley (2016) also, extend convolutions to graphs via graph diffusions. Building upon these works and the ideas of Bai et al. (2017a), Dernbach et al. (2019) incorporate quantum walks into a neural network.

In this paper, we explore the application of some quantum computing techniques in graph learning. We first summarize some basic principles relevant to our approaches in Sect. 2. Next, we present a graph learning method that is inspired by these quantum ideas. Our approach is a hybrid one that a) uses quantum walks to learn diffusions and b) utilizes the diffusions in a classical way. Finally, we present the results of our experiments.

2 Preliminaries

2.1 Bosonic Quantum Mechanics

Photonic circuits are a prime candidate for both near-term and future quantum devices. Photons are a type of boson that lead to interesting statistical and physical phenomena. Hence understanding some key aspects of bosonic quantum systems is important when considering possibly physical realization of some quantum algorithms.

An important characteristic of bosons is that two bosons of the same type are indistinguishable; and this has interesting consequences. Let us denote system state by $|\psi(x_1, x_2)\rangle$ where ψ is some function and x_1, x_2 are generalized coordinates (eg position, state etc.) of the two particles. The exchange operator E is then defined by the following action

$$E|\psi(x_1, x_2)\rangle = |\psi(x_2, x_1)\rangle$$

Informally the exchange operator swaps the coordinates/states of the individual particles in the combined system. The requirement of indistinguishability is the invariance of a multi-particle system to the exchange operator. Imagine a system having binary states $|0\rangle, |1\rangle$ and two particles that can occupy those states. The standard computational decomposition for the Hilbert space of such a system has 4 basis viz. $|00\rangle, |11\rangle, |01\rangle, |10\rangle$; corresponding to the states of each of the two particles. One can imagine that the system is described by the state $|01\rangle$. However, indistinguishability means we can permute the labels of the particles and the system is equally be described by $|10\rangle$. Note that this along with the superposition principle implies that any superposition of the these states is an equally valid description of the state. This leads to the following principle:

Symmetrization Postulate for Bosons. In a system of indistinguishable bosons, the only possible states of the system are ones that are symmetric with respect to permutations of the labels of those particles.

The Symmetrization Postulate restricts the Hilbert space of the system to lie in the completely symmetric subspace. In the specific case considered above the Hilbert space of this system is spanned by 3 states instead of the usual 4. Furthermore the postulate implies that result of any measurement of a state must project the state into the symmetric indistinguishable subspace.

Fock Space. Indistinguishability also implies that only the total number of bosons in a given state has any meaning. This makes it convenient to use an alternate basis for describing the Hilbert space of the system known as the Fock basis. It is a construction for the state space of a variable or unknown number of identical particles from the Hilbert space of a single particle. For bosons, the n -particle states are elements in the symmetric product of n uni-particle Hilbert spaces.

Creation and Annihilation Operators. Operations in the Fock space bases are written as unitary matrices of creation and annihilation operators. As its name suggests the creation (or raising) operator (denoted commonly by a) adds a particle to the state it operates on, while the annihilation (lowering) operator (a^\dagger) does the opposite.

2.2 Quantum Walks

A classical walk on a graph $G = (V, E)$ can be described as follows. At any given timestep t the walker resides at one of the nodes of the graph say u . Each node has an associated multiheaded coin with each head corresponding to an incident edge. Equivalently there is a multinomial distribution over the edges incident on the node. The walker then draws an edge according to the distribution (say $e = (u, v)$). The walker then moves to the node v which is connected to e .

Quantum walks are the quantum extension of a classical random walk. A classical walk involves a walker moving around on a graph and at any point in time its position is given by a probability distribution. A quantum walk is similar, however instead of a random process the walkers movement is governed by a sequence of unitary operations. Unlike a classical walk where the walker can only be at a node, the quantum walker can be in a superposition over all nodes in the graph.

We follow the approach of Kendon (2011) in describing quantum walks. A quantum walk involves two Hilbert spaces: the position space \mathcal{H}_v corresponding to nodes of the graph; and a coin space \mathcal{H}_c . To preserve unitarity the size of the coin space is fixed across all nodes. This can be achieved by taking the maximum degree of the nodes as the size of the space. The quantum walker's state is determined by the combined space of position and coin combinations. Instead of a coin toss, we now have a unitary operator C (called the coin operator) on the coin space describing the evolution of the coin-part of the walkers state. We also have a shift operator S . The shift operator acts as a conditional gate: depending

on the coin state it swaps the coefficients of the corresponding positions. The evolution of the entire system is given by the unitary operator $U = S(IC)$

This kind of evolution produces a behavior completely unlike that of a classical walk. The superposition of states allows walker trajectories to interfere: something that cannot happen classically. This interference can lead to a faster spreading of the walker's final position distribution. The clear effect of this can be seen in Fig. 1. These figures show the result of simulating a classical and quantum walk on a 1-d lattice for 30 steps. The final state distribution of classical walk is shown in Fig. 1a. It is clearly centered around its starting point and has a exponentially falling tail as one moves further from the start.

The quantum walk however (Fig. 1b) shows a very different picture. While the mean of the walk is still at the starting point, the distribution modes are peaks far away from the start. Such behavior allows a quantum random walker to have significantly better exploration. Inspiring from this insight Dernbach et al. (2018) proposed a version of diffusion networks based upon quantum walks. They demonstrated that using probability distribution of quantum walkers allows for far better exploration and incorporation of graph structure as compared to classical approaches.

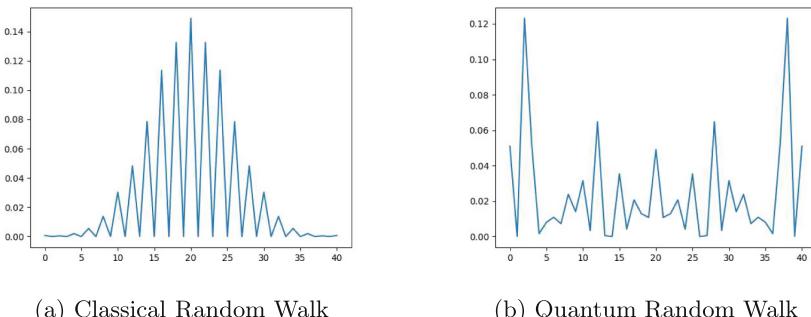


Fig. 1. Comparing classical and quantum walk

3 Bosonic Walks Networks

A natural question with respect to random walk based graph networks is whether incorporation of multiple walkers can lead to a different outcome. While multiple non-interacting classical walkers have no extra power compared to a single classical walker, the answer is different for quantum walkers (Chandrashekar and Busch 2012). A key reason for this is the symmetrization postulate referred to earlier.

3.1 Bosonic Quantum Walks

Bosonic walks can a) have unintuitive non-local correlations across walker states and b) allow for dynamics not accessible for distinguishable particles. As such even limited bosonic walks can have surprising power. For example Gamble et al. (2010) demonstrate that there are classes of non-isomorphic graphs that can be distinguished by the node distribution of multi-particle walks but not by the node distribution of a single particle quantum walk. More recently Lahini et al. (2018) have proposed a scheme for implementing high-fidelity quantum gates using a multiple bosonic quantum walkers.

We focus on a setting with multiple bosonic particles executing quantum walks on a graph. The Hilbert space of these walkers allow significantly more trajectories of the walker by allowing entangled coins and other complexities. The evolution of the walkers state in the Fock space basis is driven by the Hamiltonian H given by

$$H = A_{ij}c_i^\dagger c_j + Ec_i c_i^\dagger (c_i c_i^\dagger - 1)$$

A is the adjacency matrix of the graph and c_i, c_i^\dagger are creation and annihilation operators associated with node i . $Ec_i c_i^\dagger (c_i c_i^\dagger - 1)$ is a term which describes the interaction between the walkers and E is the interaction strength. Note that when the particles are in different nodes, the interaction term has no effect.

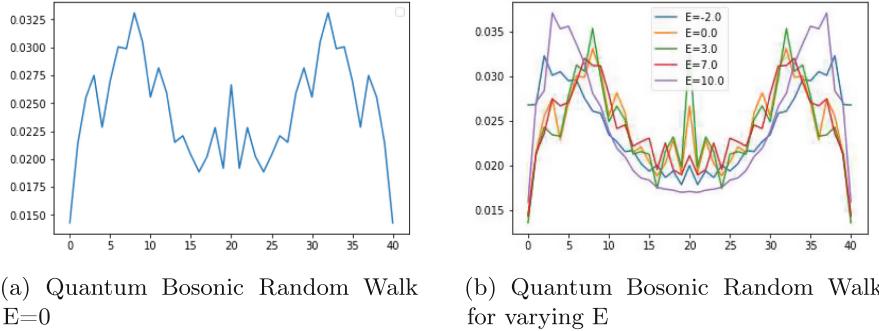


Fig. 2. Comparing quantum walks with difference interaction energy U

Similar to earlier results, we compute the probability of observing a walker across different nodes on a 1-D lattice after 30 steps in the case when $E = 0$. This is plotted in Fig. 2a, where we can notice easy differences between this plot and the plot of the single particle quantum walk discussed earlier. Next we plot the probability of observing a walker for different values of E in Fig. 2b. Note the probability distribution changes significantly as the interaction strength changes.

3.2 Bosonic Quantum Walk Neural Networks

A Bosonic Quantum Walker Network (QWB) is the natural extension of the Dernbach et al. (2018) model using bosonic quantum walkers. For our description below, we will follow a similar presentation. The key idea behind a quantum walk neural network is to use the walker's distribution over the nodes of a graph to construct a diffusion matrix, which is then utilized to aggregate information from the nodes. At each time step, we simulate the dynamics of the walker using the coin operator, C , to modify the spin state of the walkers ψ according to $C(t)\psi_t \rightarrow \psi_{t+1}$. The coin operator need not be static and can depend on both time and node features. This is followed by the shift operator, which moves the walker to a neighbouring node depending upon the walker spin state. The walker dynamics induces a probability distribution of the walker over the graph (written as a probability matrix P). Next, this matrix P is used to diffuse the node level features across the graph: $\hat{X} = PX$. These diffused features are the output of a single quantum diffusion layer. These features can then be used either as input for a second diffusion layer; or for final prediction. All of these operations are differentiable, and hence we can use backpropagation to compute the gradient of the loss with respect to all the model parameters (especially parameters of the coin matrix) (Fig. 3).

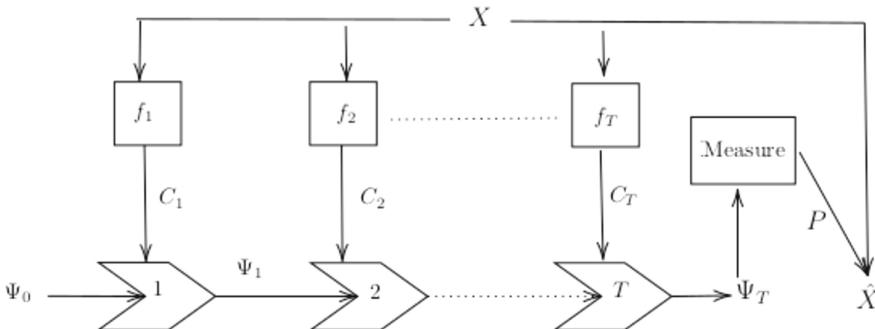


Fig. 3. Quantum Walk Neural Network Schematic: The feature matrix X is used to produce the coin operators C_i used in each step i . The superposition ψ evolves after each step. The final layer diffuses X using measured probabilities P to compute \hat{X}

Note that the description till here is independent of how the walkers behave. In fact, walkers can behave completely classically, in which case the behaviour is identical to the Diffusion Convolution model of Atwood and Towsley (2016). For a quantum walk network, the walk dynamics are governed by quantum evolution. The induced probability matrix is the one determined by the measurement of the walker's node-state. We compute P induced by a bosonic multi-particle quantum walk on the graph in the bosonic quantum walk model. This can be achieved via evolving single-particle walkers and then performing a projection into the indistinguishable space by performing repeating sums over identical labels.

Since a k -particle quantum walk naturally produces a symmetric superposition over k -tuples of nodes, we can, in principle, extend P from a distribution over nodes to a distribution over node pairs or even higher orders. In our experiments, however, we do not use such higher-order features. Instead, we compute P from the probability of observing *a single* boson at the given nodes.

4 Experiments

Datasets. We experiment with commonly used graph datasets: QM7, which is a regression task and MUTAG, NCI1, and Enzymes (which are classification problems). MUTAG Debnath et al. (1991) is a dataset of 188 mutagenic aromatic and heteroaromatic nitro compounds that are classified as either mutagenic or not. NCI1 Wale and Karypis (2006) consists of 4110 graphs representing two sets of chemical compounds screened for activity against non-small cell lung cancer. For both these datasets, each graph represents a molecule, with nodes representing atoms and edges representing bonds between atoms. Each node has an associated label that corresponds to its atomic number. Enzymes Borgwardt et al. (2005) is a dataset of 600 molecules where the task is to classify each enzyme into one of six classes. The QM7 dataset Rupp et al. (2012); Blum and Reymond (2009) is a collection of 7165 molecules, each containing up to 23 atoms. The goal of the task is to predict the atomization energy of each molecule.

Experimental Details. We include as baselines two classical methods (DCNN, GCN). DCNN refers to the diffusion convolutional network of Atwood and Towsley (2016), while GCN is the graph convolution architecture of Kipf and Welling (2017). QWNN is the quantum walk based model presented in Dernbach et al. (2019). QWB2 is our two-particle bosonic extension of the QWNN model. The metric used for classification tasks (NCI, MUTAG, Enzymes) is accuracy (so higher the better) while the one used for QM7 is mean prediction error (so lower the better).

Classical simulation of quantum walks has poor scaling properties. Simulating a k -particle quantum walk scales exponentially in k . As such in our experiments we restricted ourselves to graphs of size less than 70 and had to reduce sizes of feature embeddings. Furthermore to keep comparison fair in terms of feature size between classical and quantum models, we applied the same restriction to the classical models as well.

Results. Table 1a reports the classification performance of different models. We see that QWB2 outperforms other models, especially on Enzymes where the gain is substantial. QWB2 also outperforms QWNN and other classic approaches on MUTAG and NCI datasets. Regression results on QM7 are presented in Table 1b. The basic trend of QWB2 outperforming other approaches remains, though the performance differences are comparatively smaller. Overall QWB2 seems to outperform all other models including its single-walker counterpart QWNN.

Table 1. Results of different models on graph classification and regression tasks

Model	Enzymes	MUTAG	NCI
GCN	31.4	87.4	69.6
DCNN	27.9	89.1	69.1
QWNN	33.6	88.4	73.6
QWB2	40.2	90.0	76.7

(a) Classification tasks

Model	MSE	MAE
GCN	17.5	12.4
DCNN	11.9	8.6
QWNN	10.9	8.4
QWB2	9.2	7.9

(b) Atomization energy prediction on QM7

5 Related Work

Graph Neural Networks. Early graph neural networks (GNN) (Gori et al. 2005; Scarselli et al. 2008) used recursive architectures to encode graphs into finite-dimensional vectors. Since then has been tremendous progress in learning representations of graphs. Convolutional neural networks (Bruna et al. 2013; Defferrard et al. 2016; Kipf and Welling 2017) borrow ideas from graph Laplacians (Cvetkovic et al. 1998) for processing graphs signals. Atwood and Towsley (2016), on the other hand, proposed a spatial approach relying on random walks. Gilmer et al. (2017) proposed a general approach for learning on graphs via message passing between nodes of the graphs. All the other mentioned works can be interpreted as a restricted version of that approach.

Quantum Models. There is a rich literature exploring quantum walks beginning with works of Ambainis et al. (2001) and Aharonov et al. (2001). A generalization of discrete walks for an arbitrary number of walkers was studied by Rohde et al. (2011). Subsequently, multiple works have developed graph kernels based on the quantum walks (Rossi et al. 2013; Bai et al. 2013, 2017b). Quantum walks have also been shown to provide a model for universal computation Childs (2009). They have been explored for algorithmic applications Childs and Eisenberg (2003); Qiang et al. (2012) and quantum simulation Berry et al. (2005). While there have been multiple proposals of quantum neural networks over the years Gupta and Zia (2001); Biamonte et al. (2017); there has been not much work done on quantum learning techniques for graphs. Our work derives from the recent work of Dernbach et al. (2018), which proposed a quantum version of graph diffusion networks.

6 Conclusion

In this work we have tried to incorporate multi-particle bosonic walks on graphs. Unlike simple QWNN, this approach allows for learning significantly more powerful and complex graph diffusions. This benefit is clear across both regression and classification tasks. A future research direction would be to find ways to constrain the multiple walkers such that the simulation becomes more feasible.

References

- Aharanov, D., Ambainis, A., Kempe, J., Vazirani, U.: Quantum walks on graphs. In: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, pp. 50–59 (2001)
- Ambainis, A., Bach, E., Nayak, A., Vishwanath, A., Watrous, J.: One-dimensional quantum walks. In: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, pp. 37–49 (2001)
- Atwood, J., Towsley, D.F.: Diffusion-convolutional neural networks. In: NIPS (2016)
- Bai, L., Hancock, E.R., Torsello, A., Rossi, L.: A quantum Jensen-Shannon graph Kernel using the continuous-time quantum walk. In: International Workshop on Graph-Based Representations in Pattern Recognition, pp. 121–131. Springer (2013). https://doi.org/10.1007/978-3-642-38221-5_13
- Bai, L., et al.: Quantum Kernels for unattributed graphs using discrete-time quantum walks. *Patt. Recogn. Lett.* **87**, 02 (2017)
- Bai, L., et al.: Quantum Kernels for unattributed graphs using discrete-time quantum walks. *Patt. Recogn. Lett.* **87**, 96–103 (2017)
- Berry, D., Ahokas, G., Cleve, R., Sanders, B.: Efficient quantum algorithms for simulating sparse Hamiltonians. *Commun. Math. Phys.* **270**, 09 (2005). <https://doi.org/10.1007/s00220-006-0150-x>
- Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. *Nature* **549**(7671), 195–202 (2017)
- Blum, L.C., Reymond, J.-L.: 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. *J. Am. Chem. Soc.* **131**(25), 8732–8733 (2009)
- Bonetta, R., Valentino, G.: Machine learning techniques for protein function prediction. *Prot. Struct. Funct. Bioinf.* **88**, 10 (2019)
- Borgwardt, K.M., Ong, C.S., Schönauer, S., Vishwanathan, S., Smola, A.J., Kriegel, H.-P.: Protein function prediction via graph Kernels. *Bioinformatics* **21**(suppl_1), i47–i56 (2005)
- Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. arXiv preprint <arXiv:1312.6203> (2013)
- Callut, J., Francois, K., Saerens, M., Dupont, P.: Classification in graphs using discriminative random walks. In: International Workshop on Mining and Learning with Graphs, July 2008
- Chandrashekar, C.M., Busch, T.: Quantum walk on distinguishable non-interacting many-particles and indistinguishable two-particle. *Quantum Inf. Process.* **11**(5), 1287–1299 (2012)
- Childs, A.M.: Universal computation by quantum walk. *Phys. Rev. Lett.* **102**, 180501 (2009)
- Childs, A.M., Eisenberg, J.M.: Quantum algorithms for subset finding. *Quantum Inf. Comput.* **5**, 593–604 (2003)
- Crossman, L.C.: Leveraging deep learning to simulate coronavirus spike proteins has the potential to predict future zoonotic sequences. bioRxiv (2020)
- Cvetkovic, D., Doob, M., Sachs, H.: Spectra of Graphs: Theory and Applications, 3rd rev. enl. ed (1998)
- Debnath, A.K., Lopez de Compadre, R.L., Debnath, G., Shusterman, A.J., Hansch, C.: Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *J. Med. Chem.* **34**(2), 786–797 (1991)

- Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: NIPS (2016)
- Dernbach, S., Mohseni-Kabir, A., Pal, S., Towsley, D., Gepner, M.: Quantum walk inspired neural networks for graph-structured data. arXiv: Quantum Physics (2018)
- Dernbach, S., Mohseni-Kabir, A., Pal, S., Gepner, M., Towsley, D.: Quantum walk neural networks with feature dependent coins. Appl. Netw. Sci. **4**(1), 1–16 (2019). <https://doi.org/10.1007/s41109-019-0188-2>
- Dobson, P., Doig, A.: Distinguishing enzyme structures from non-enzymes without alignments. J. Mol. Biol. **330**, 771–783 (2003)
- Duvenaud, D.K., et al.: Convolutional networks on graphs for learning molecular fingerprints. In: Advances in Neural Information Processing Systems **28**, 2224–2232 (2015)
- Gamble, J.K., Friesen, M., Zhou, D., Joynt, R., Coppersmith, S.N.: Two-particle quantum walks applied to the graph isomorphism problem (2010)
- Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. arXiv preprint arXiv:1704.01212 (2017)
- Gori, M., Monfardini, G., Scarselli, F.: A new model for learning in graph domains. In: Proceedings. 2005 IEEE International Joint Conference on Neural Networks 2005, vol. 2, pp. 729–734. IEEE (2005)
- Gupta, S., Zia, R.: Quantum neural networks. J. Comput. Syst. Sci. **63**(3), 355–383 (2001)
- Gärtner, T., Flach, P., Wrobel, S.: On graph kernels: Hardness results and efficient alternatives, vol. 129–143, pp. 129–143, January 2003. https://doi.org/10.1007/978-3-540-45167-9_11
- John, P.S., Guan, Y., Kim, Y., Kim, S., Paton, R.: Prediction of Homolytic Bond Dissociation Enthalpies for Organic Molecules at Near Chemical Accuracy with Sub-Second Computational Cost, November 2019
- Kendon, V.: Quantum walks on general graphs. Int. J. Quantum Inf. **04**, 11 (2011)
- Kipf, T., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
- Kondor, R., Borgwardt, K.M.: The skew spectrum of graphs. In: Proceedings of the 25th International Conference on Machine Learning, ICML 2008, pp. 496–503 (2008)
- Kondor, R., Shervashidze, N., Borgwardt, K.: The graphlet spectrum, vol. 382, p. 67, January 2009. <https://doi.org/10.1145/1553374.1553443>
- Lahini, Y., Steinbrecher, G.R., Bookatz, A.D., Englund, D.: Quantum logic using correlated one-dimensional quantum walks. NPJ Quantum Inf. **4**(1), 1–7 (2018). ISSN 2056–6387
- Qiang, X., Yang, X., Wu, J., Zhu, X.: **45**(4), 045305 (2012)
- Rohde, P.P., Schreiber, A., Štefaňák, M., Jex, I., Silberhorn, C.: Multi-walker discrete time quantum walks on arbitrary graphs, their properties and their photonic implementation. New J. Phys. **13**(1), 013001 (2011)
- Rossi, L., Torsello, A., Hancock, E.R.: A continuous-time quantum walk kernel for unattributed graphs. In: International Workshop on Graph-Based Representations in Pattern Recognition, pp. 101–110. Springer (2013). https://doi.org/10.1007/978-3-642-38221-5_11
- Rupp, M., Tkatchenko, A., Müller, K.-R., von Lilienfeld, O.A.: Fast and accurate modeling of molecular atomization energies with machine learning. Phys. Rev. Lett. **108**, 058301 (2012)

- Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. *IEEE Trans. Neural Netw.* **20**(1), 61–80 (2008)
- Vishwanathan, S., Schraudolph, N.N., Kondor, R., Borgwardt, K.M.: Graph Kernels. *J. Mach. Learn. Res.* **11**(40), 1201–1242 (2010)
- Wale, N., Karypis, G.: Comparison of descriptor spaces for chemical compound retrieval and classification. In: Sixth International Conference on Data Mining (ICDM-06). IEEE (2006)

Machine Learning and Networks



Multiple Role Discovery in Complex Networks

Shu Liu^{1(✉)}, Fujio Toriumi¹, Mao Nishiguchi¹, and Shohei Usui²

¹ The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8654, Japan

{liu,nishiguchi}@torilab.net, tori@sys.t.u-tokyo.ac.jp

² Sansan Inc., Tokyo, Japan

usui@sansan.com

Abstract. The role of a node in complex networks is the aggregation of structural features and functions. Role discovery is the field of mining the proper roles, and many methods have been proposed. Those methods mainly focus on discovering a single role for each node. However, in real-world networks, a node may have multiple roles. Therefore, we propose a multiple-role discovery framework by extending the single-role discovery framework. Furthermore, we also suggest a way to assign sub-networks divided by community extraction methods to the source network and the validation network to select pre-labeling nodes, which is a significant challenge for multiple-role discovery in real-world networks. To evaluate the accuracy of the proposed method, we conduct computational experiments for multiple-role discovery of the real-world Wikipedia network and Blogcatalog network. We show that the proposed method achieves higher accuracy and more stable results than conventional methods used for comparison.

Keywords: Role discovery · Multi-label classification · Transfer learning

1 Introduction

In complex networks, the role of a node is to aggregate structural features and functions. In the context of network topology, the roles of nodes include star-center, star-periphery, clique-member, etc. [1], while in the context of directed networks, source and sink should be considered [2]. Role discovery, the study of assigning roles to nodes, has evolved to provide a more intuitive understanding of complex network structures that are difficult for humans to grasp. It is not only crucial for graph mining and exploratory analysis but also useful in many practical applications.

In real-world networks, a single node may have multiple roles simultaneously, depending on the construction method. In temporal networks, the roles of nodes may change over time, and depending on the granularity of the partitioning, multiple roles can be observed within a single time slice [1]. For example, in a

human-relationship network in which people are nodes and connections between people are links, a node that plays the role of an office worker during weekday working hours and a father during other hours can be said to play the roles of an office worker and a father from a long-term perspective. In addition, in a word co-occurrence network, if part of speech is used as a role label, a word with multiple parts of speech will have multiple roles. The phenomenon of multiple roles of nodes exists generally, and not only in temporal networks. Although existing research recognizes the multiple-role phenomenon, to our best knowledge, no specific approach to the discovery of multiple roles has been presented [1].

This study aims to apply the multi-label classification approach to role discovery in complex networks and to predict multiple roles with high accuracy. First, we propose a framework for multiple-role discovery by extending the previous method (referred to below as STV method) [3]. The STV method is a supervised learning approach that transfers single-role knowledge from a pre-labeled network (source network) to a network where the roles are unknown (target network), using domain adversarial learning, and then selects optimal hyperparameters from a grid search using another pre-labeled network (validation network). Second, we suggest a way to select appropriate source and validation networks when applying the proposed method to real-world networks using community extraction. The accuracy of the proposed method is verified using two real-world networks, namely the Wikipedia network and the Blogcatalog network [4].

2 Related Work

2.1 Role Discovery

Role discovery is the task of partitioning nodes into groups of nodes, or role classes, that have a certain equivalence. Conventional methods for role discovery are mainly unsupervised learning and supervised learning approaches.

An unsupervised learning approach usually obtains a vector representation of nodes from the network structure, clusters nodes according to the vector representation, and assigns a role label to each cluster. The most famous such approach is the block model [5–8]. The block model is typically formulated as an optimization problem using a well-formed objective function for a constructed role-interaction graph. In this role interaction graph, nodes with the same role are aggregated and represented as blocks (nodes), while the edges between the blocks show the interaction between the roles. Some models assign each node to one of several roles (or blocks), while Airoldi et al. [9] proposed a mixed-membership stochastic block model that relaxes the assumption of nodes belonging to only a single role. This model is characterized by a soft assignment method that assigns a node to a probability vector belonging to multiple roles. Some methods use a form of similarity between the rows of the adjacency matrix to calculate the roles [10,11]. These methods mainly involve calculating the similarity (or distance) between each pair of rows of the adjacency matrix to gain the similarity matrix and then clustering the nodes using that similarity matrix. There are also

spectral methods that use a subset of the eigenvectors of the adjacency matrix (or similarity matrix) to derive the roles [11].

Unsupervised learning methods have the disadvantage of requiring subjective guesswork when assigning specific roles to individual clusters, making the roles less interpretable and the task of checking the validity of the assigned roles challenging.

On the other hand, the supervised learning approach trains the learning model through predefined roles, role assignment criteria, or positive examples to predict the roles of the test data. In the supervised learning approach, the roles are predetermined as needed, which improves interpretability, and the accuracy of the predictions can be maintained at a certain level by learning from positive examples. The STV method of role discovery for nodes using transfer learning [3] is an example of a supervised learning approach, where role knowledge from a pre-labeled network (source network) is transferred to a network where the roles are unknown (target network).

Both approaches are aimed at discovering a single role of a node but are not capable of multiple-role discovery. An approach mentioned above that is closer to discovering multiple roles assigns a probability vector (where the sum of the vector elements equals 1) belonging to multiple roles to a node instead of assigning a single role label to a node, known as the soft assignment method. The soft assignment method is inadequate for multiple-role discovery because it can only provide a rough trend of the multiple roles of a node.

2.2 Multi-label Classification

The multi-label classification task is a type of supervised learning in which the output data corresponding to one input instance is a vector of multiple binary values (0 and 1). The number of elements in the vector corresponds to the number of labels to be classified, where 0 means that the label is negative and 1 means that the label is positive. Compared to a binary classification task where the output data is a binary scalar, or a multi-class classification task where the output data is one number representing one of the limited categories, multi-label classification is a more challenging task because it is more susceptible to problems such as high dimensionality and label imbalance [12].

Multi-label classification can be achieved through various approaches such as ensemble of classifiers, data transformation, and method adaptation.

Binary Relevance (BR) is a classifier ensemble method. Multiple binary classifiers are pre-trained for each label. The output of multiple binary classifiers is concatenated as the output vector for the test data instance. The binary classifier can be any binary classification algorithm. The advantage is that each label can be trained independently and in parallel with the others, while the disadvantage is that correlations between labels are ignored. Moreover, the prediction result can be negative for all labels.

Label Powerset (LP) is a data transformation approach that transforms multi-label classification tasks into multi-class classification tasks. Specifically, it creates a new class label indicating each different combination of original labels

for all data instances, trains a multi-class classifier to predict the new class labels, and then retrieves the original label combination. Unlike BR, LP preserves the correlation between labels. Furthermore, LP methods can apply the know-how of multi-class classification methods.

MLkNN is an adaptation method that adapts the kNN method to multi-label classification [13]. It predicts the label vector of the corresponding instance based on the label vectors of the k -nearest neighbors. Unlike the k -nearest neighbor method, it generates a model that includes prior and conditional probabilities for each label.

3 Previous Method (STV Method) [3]

The pipeline of the STV method is shown in Fig. 1. The gray arrows represent struc2vec, which acquires the node representation (shown in the green box) of the networks on the left side. d is the number of dimensions for the node representation; $|V_s|$, $|V_t|$, and $|V_v|$ are the number of nodes in source, target, and validation network, respectively. The node representation is used as the input of two neural networks, the Discriminator f_{θ^D} (blue box) and the Role model f_{θ^R} (orange box). Each neural network updates its weights according to the specific loss function in the training step, represented by the thin black arrow inside the box from right to left. Also, the node representation is modified to be more apparent for role prediction and more difficult for domain discrimination by a gradient reversal layer [14], represented by the thin black arrows pointed to the green box.

3.1 Acquisition of Node Representation

struc2vec is a method designed to obtain distributed representations so that structurally similar nodes are placed close to each other in the latent space. It is considered suitable for the role discovery of nodes because it does not depend on their locations in the network.

3.2 Domain Adversarial Learning

The differences in scale and other factors between the source and target networks might influence the distributed representation of nodes obtained by struc2vec to form different distributions in each domain, which may lead to poor performance of role estimation in the target network. Therefore, in the STV method, a domain adversarial learning framework is proposed to mitigate the differences in the domains of both networks. The domain adversarial learning framework includes the Discriminator shown in the upper-right area of Fig. 1 and the Role model shown in the lower-right area. The goal of the Discriminator is to judge whether a node belongs to the source network or the target network. The input of the Discriminator is the distributed representation of the node, and the output is a confidence score between 0 and 1. If a node approaches 0 (1, respectively), it

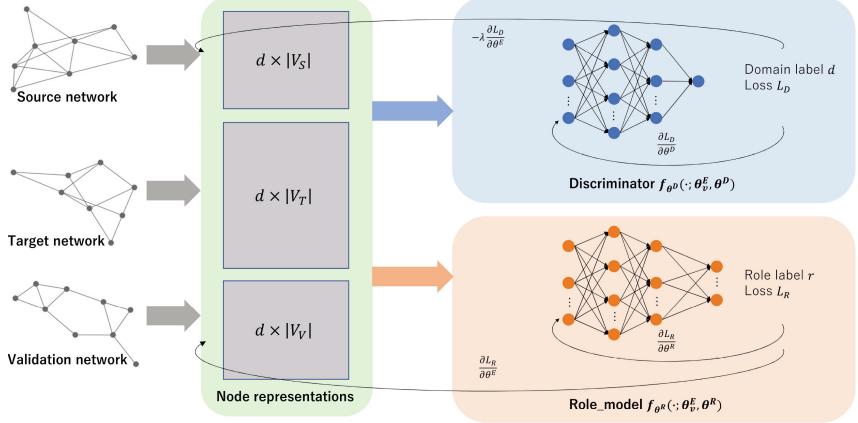


Fig. 1. Pipeline of framework for single-role discovery [3]

is judged to belong to the source (target, respectively) network. The goals of the Role model are to predict a single role label for a node and to fool the Discriminator using a gradient reversal layer [14]. Notably, the Role model consists of a 3-layer neural network, where the output layer is a fully connected layer with as many neurons as the number of roles. Since the Role model for single-role discovery is a multi-class classification task, the active function of the output layer is the LogSoftmax function, which is given by the following equation.

$$\text{LogSoftmax}(x_i) = \log \left(\frac{\exp(x_i)}{\sum_{j=1}^r \exp(x_j)} \right). \quad (1)$$

x_i is the output of the neuron corresponding to i th role. The role label corresponding to the neuron with the highest output among the r neurons is the predicted role of the corresponding instance.

Following the domain adversarial learning framework, a mini-max game is applied. Thus, by training the Role model and Discriminator alternately, the nodes' distributed representations in both networks are expected to become similar. After the learning is converged, the nodes' distributed representation in the target network is used as input to the Role model to predict the corresponding single-role label.

3.3 Model Selection Using Validation Network

The STV method enables us to approach model selection as ordinary supervised learning by using a validation network. It treats the pre-labeled validation network the same as the target network and predicts the nodes' role in the validation network against those in the target network. By comparing them with the ground truth labels, the model with the maximum accuracy can be chosen,

and thus the most confident prediction for the target network can be selected. This is based on the assumption that when an appropriate validation network is applied, the model with the highest accuracy in the validation domain will also have the highest accuracy in the target domain.

4 Proposed Method

4.1 Multi-label Classification in Neural Networks

In this research, we make the following improvements to the Role model of the STV method to enable multi-label classification for multiple-role discovery.

Modification of Output Layer Structure. To allow multi-label classification of the Role model, we changed the active function of the output layer from Eq. 1 to the sigmoid function shown below.

$$\text{Sigmoid}(x_i) = \frac{1}{1 + e^{-x_i}}. \quad (2)$$

The sigmoid function can map real numbers to the interval $(0, 1)$ and is often used as the active function for the output layer of binary classification. In multi-label classification, the output of r neurons corresponding to r labels is mapped to the interval $[0, 1]$. The output presents the corresponding label as positive or negative if it is greater or less than the threshold, respectively.

Selecting BP-MLL as Loss Function. The loss function of a neural network is crucial, and it significantly impacts the efficiency and accuracy of training. There has been much research on loss functions for multi-label classification tasks [15, 16]. In this study, we adopt a loss function called BP-MLL [15]. BP-MLL is a loss function based on binary cross-entropy, considering that each instance contains multiple labels. BP-MLL imposes a penalty when it predicts a label unrelated to an instance as a related label. As shown in the following equation, the larger the difference between the relevant label's output and an irrelevant label's output, the better:

$$E = \sum_{i=1}^m E_i = \sum_{i=1}^m \frac{1}{|Y_i||\bar{Y}_i|} \sum_{(k,l) \in Y_i \times \bar{Y}_i} \exp(-(c_k^i - c_l^i)), \quad (3)$$

where m is the number of training instances and Y_i is the label set corresponding to the i th instance; \bar{Y}_i is the complement of Y_i , which is a set of labels not associated with the instance i ; c_k^i is the output of the neuron corresponding to the label k of the instance i .

For each instance, BP-MLL sums up the error of all pairs of the outputs of the associated labels and the unassociated labels. For each learning epoch, the total loss is the sum of all instances. Thus, the output values of the associated

label set will be higher, and the output values of the unassociated label set will be lower. Therefore, BP-MLL is expected to achieve high accuracy for multi-label classification. In contrast to binary cross-entropy, which penalizes neurons by comparing their outputs to a threshold value, BP-MLL also considers the outputs between neurons, and thus it can perform multi-label classification with high accuracy.

4.2 Creation of Training Data Using Community Extraction

In most cases, roles in real-world networks are diverse and case-by-case. Moreover, in the case of multiple roles, the roles themselves and the co-occurrence of the roles are diverse. Therefore, the selection of the source network and the validation network is critical to applying the proposed method. However, from our empirical experiments, inappropriate selection of the two networks may cause selection bias, which damages accuracy. To tackle this problem, we propose a method for creating training data using community extraction. Community extraction is a task that divides nodes into groups that are tightly connected internally. The community extraction algorithm called the Leiden method [17] is based on the Louvain method [18] and guarantees that the communities are always connected.

We use the Leiden method to create training data for multiple-role discovery in real-world networks because struc2vec is expected to obtain a proper distributed representation of connected networks for the following procedures. Specifically, we randomly select two subnetworks with appropriate size from the real-world network divided by the community extraction method, label them with multiple roles, and assign one to the source network and the other to the validation network. Other subnetworks or the entire network can be regarded as the target network. Finally, the proposed method is applied to the target network to predict the multiple roles. Therefore, multiple-role discovery can be achieved in situations where roles in real-world networks are diverse and case-by-case.

5 Evaluation Experiments on Real-World Networks

5.1 Comparison Method

The comparison methods used in this study are BR (GS), LP (GS), and MLkNN, which are introduced above in Related Work. For each comparison method, we use the distributed representation of the source network obtained from struc2vec as test data, the distributed representation of the validation network as validation data, and predict role labels from the distributed representation of the target network. GS indicates that grid search is applied.

5.2 Variations of Proposed Method

We consider the following four variations of the proposed method according to the presence or absence of the Discriminator and the type of Discriminator.

BPMLL: No Discriminator is used. In other words, the domain adversarial learning framework is not applied.

BPMLL_SD2: Incorporates one Discriminator for binary classification, the same as the proposed approach in the STV method. Discriminator identifies whether the distributed representation is from the source network or the target and validation networks.

BPMLL_SD3: Incorporates one multi-level classification Discriminator, which has three output neurons corresponding to the source, target, and validation networks.

BPMLL_DD: A double Discriminator scheme that consists of two binary Discriminators. One discriminates between the source and target domains, and the other discriminates between the source and validation domains.

5.3 Evaluation Metrics

In this study, the F-measure is used to evaluate accuracy.

$$F\text{-measure} = 2 * \frac{Precision * Recall}{Precision + Recall}, \quad (4)$$

$$Precision = \frac{1}{m} \sum_{i=1}^m \frac{|Y_i \cap Z_i|}{|Z_i|}, \quad Recall = \frac{1}{m} \sum_{i=1}^m \frac{|Y_i \cap Z_i|}{|Y_i|}. \quad (5)$$

The true label matrix is denoted by $Y \in \{0, 1\}^{m \times r}$, and the output of the classifier is denoted by $Z \in \{0, 1\}^{m \times r}$. r is the number of roles. F-measure is an instance-based evaluation metric for multi-label classification, different from the conventional Macro-F1, Micro-F1, etc.

5.4 Wikipedia Network

The Wikipedia network is a word co-occurrence network using a portion of Wikipedia as its corpus, with 4,777 nodes, 184,812 edges, and 40 different node role labels (part-of-speech labels for words) used for discovery [4]. Community extraction was applied on the entire network (Wiki_all), and three sub-networks (Wiki_c1, Wiki_c2, and Wiki_c3) were randomly selected. In the three sub-networks, 18 mutual role labels were used as prediction targets. The three sub-networks were randomly assigned as the source network, target network, and validation network, and six combinations of them were formed as six datasets. Table 1 and Table 2 summarize the statistics of the communities and the combinations composing the six datasets.

Ten experiments were conducted on each dataset, and the average F-measure, Precision, and Recall results are given in Table 3. The boxplots of F-measure results on whole datasets are shown in Fig. 2.

Table 1. Statistics of Wikipedia network and communities

Name	# Node	# Edge	# Role
Wiki_all	4,777	184,812	40
Wiki_c1	958	7,946	18
Wiki_c2	1,633	8,550	18
Wiki_c3	442	1,982	18

Table 2. Wikipedia datasets

No	Source	Target	Validation
Wiki-1	Wiki_c1	Wiki_c2	Wiki_c3
Wiki-2	Wiki_c1	Wiki_c3	Wiki_c2
Wiki-3	Wiki_c2	Wiki_c1	Wiki_c3
Wiki-4	Wiki_c2	Wiki_c3	Wiki_c1
Wiki-5	Wiki_c3	Wiki_c1	Wiki_c2
Wiki-6	Wiki_c3	Wiki_c2	Wiki_c1

Table 3. Results on Wikipedia datasets

Dataset	Metric	BPMILL	BPMILL_SD2	BPMILL_SD3	BPMILL_DD	BR(GS)	LP(GS)	MLkNN
Wiki-1	F-measure	0.416	0.506	0.495	0.470	0.004	0.379	0.101
	Precision	0.321	0.448	0.453	0.415	0.003	0.346	0.099
	Recall	0.610	0.587	0.560	0.563	0.007	0.420	0.103
Wiki-2	F-measure	0.353	0.402	0.400	0.370	0.009	0.269	0.194
	Precision	0.276	0.438	0.424	0.366	0.009	0.305	0.201
	Recall	0.490	0.372	0.382	0.382	0.009	0.241	0.187
Wiki-3	F-measure	0.281	0.307	0.304	0.299	0.294	0.297	0.226
	Precision	0.212	0.271	0.273	0.284	0.343	0.349	0.214
	Recall	0.431	0.355	0.351	0.324	0.256	0.259	0.240
Wiki-4	F-measure	0.398	0.396	0.410	0.403	0.388	0.387	0.317
	Precision	0.317	0.353	0.406	0.339	0.442	0.437	0.305
	Recall	0.552	0.457	0.421	0.510	0.346	0.347	0.331
Wiki-5	F-measure	0.288	0.299	0.298	0.294	0.112	0.252	0.206
	Precision	0.257	0.337	0.330	0.312	0.134	0.293	0.207
	Recall	0.332	0.268	0.273	0.283	0.097	0.221	0.204
Wiki-6	F-measure	0.480	0.509	0.495	0.491	0.148	0.383	0.339
	Precision	0.388	0.478	0.480	0.470	0.170	0.429	0.347
	Recall	0.632	0.551	0.517	0.527	0.131	0.346	0.331

The variations of the proposed methods with Discriminator (BPMILL_SD2, BPMILL_SD3, and BPMILL_DD) have higher average F-measure results than do the comparison methods. In the case of Wiki-2 shown in Fig. 2b, BPMILL_SD2 and BPMILL_SD3 achieved 1.3 times higher accuracy than the F-measure of LP, the comparison method with the highest accuracy. BPMILL achieved higher accuracy than the comparison method in all cases except Wiki-1 and Wiki-3. However, the average F-measure was lower in all cases compared to the proposed method with the Discriminator. We believe this indicates that the domain adversarial learning framework of the proposed method was able to learn more precise classification boundaries by making the distributed representation of nodes in the source network similar to those in the target and validation networks.

In comparing the variations with the Discriminator (BPMILL_SD2, BPMILL_SD3, and BPMILL_DD), we found that the mean, variance, and number of outliers change in each combination pattern. However, there is no significant

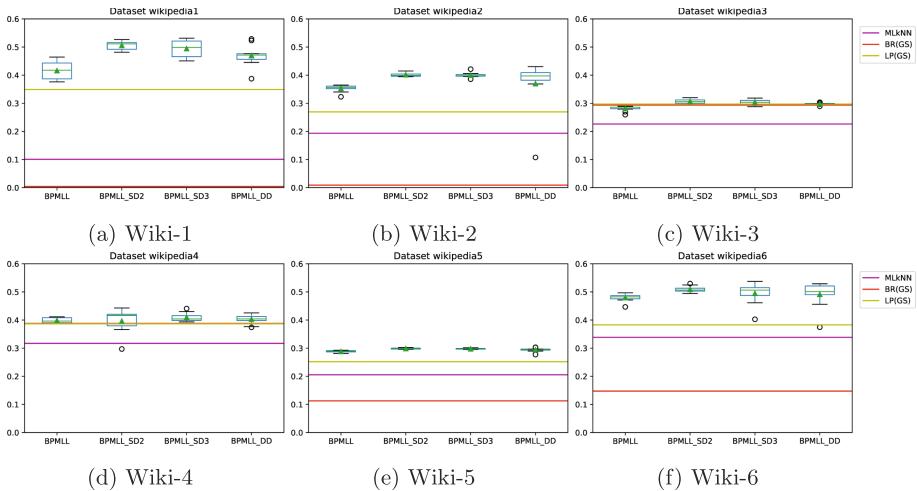


Fig. 2. Boxplots for F-measure results of Wikipedia network experiment

overall trend, which indicates that the experiment setting is not precise enough to show the predominance of any variation. We leave this as a future task.

5.5 Blogcatalog Network

The Blogcatalog network is an interaction network among bloggers in online blogging services, where the bloggers' hobbies play the role of nodes [4]. The training dataset was constructed by community extraction, similar to Wikipedia datasets. In the three sub-networks, we used 38 mutual role labels as prediction targets. Table 4 and Table 5 summarize the statistics of the communities and the combinations of the six datasets.

Ten experiments were conducted on each of the six datasets, and we show the average F-measure, Precision, and Recall results in Table 6. The boxplots of F-measure results on entire datasets are shown in Fig. 3.

Table 4. Statistics of Blogcatalog network and communities

Name	# Node	# Edge	# Role
Blog_all	10,312	333,983	39
Blog_c1	1,000	9,812	38
Blog_c2	769	2,015	38
Blog_c3	1,000	9,275	38

Table 5. Blogcatalog datasets

No.	Source	Target	Validation
Blog-1	Blog_c1	Blog_c2	Blog_c3
Blog-2	Blog_c1	Blog_c3	Blog_c2
Blog-3	Blog_c2	Blog_c1	Blog_c3
Blog-4	Blog_c2	Blog_c3	Blog_c1
Blog-5	Blog_c3	Blog_c1	Blog_c2
Blog-6	Blog_c3	Blog_c2	Blog_c1

All variations of the proposed method have higher mean F-measure results compared to the comparison methods. In the case of Blog-4 shown in Fig. 3d, the accuracy of the proposed method is nearly twice as high as the F-measure of the best comparison method, LP. The proposed method variations with the Discriminator (BPMLL_SD2, BPMLL_SD3, and BPMLL_DD) achieved higher accuracy than BPMLL without the Discriminator in Blog-1, Blog-2, Blog-3, and Blog-5. This indicates that more stable prediction can be achieved through adversarial learning of the Discriminator and Role model.

In comparing the proposed method's variations with the Discriminator, just as with the Wikipedia experiment's results, the mean value of each combination pattern is either high or low, and no general superiority or inferiority relationship appears. On the other hand, BPMLL_DD seems to have a larger variance and more outliers. This means that the Role model has more freedom to change the distributed representation of the nodes to fool the two discriminators simultaneously. This makes it difficult to converge learning as shown in Blog-6; it also increases the possibility of reaching a better equilibrium state, as shown in Blog-4.

Table 6. Results on Blogcatalog datasets

Dataset	Metric	BPMLL	BPMLL_SD2	BPMLL_SD3	BPMLL_DD	BR(GS)	LP(GS)	MLkNN
Blog-1	F-measure	0.119	0.132	0.137	0.128	—	0.088	0.058
	Precision	0.069	0.098	0.093	0.093	0.000	0.101	0.063
	Recall	0.431	0.224	0.277	0.256	0.000	0.079	0.054
Blog-2	F-measure	0.119	0.126	0.127	0.123	—	0.084	0.066
	Precision	0.072	0.091	0.092	0.091	0.000	0.096	0.069
	Recall	0.355	0.207	0.207	0.204	0.000	0.075	0.064
Blog-3	F-measure	0.131	0.155	0.155	0.156	—	0.097	0.072
	Precision	0.075	0.107	0.106	0.104	0.000	0.106	0.070
	Recall	0.521	0.291	0.294	0.318	0.000	0.090	0.075
Blog-4	F-measure	0.132	0.120	0.109	0.122	—	0.065	0.057
	Precision	0.081	0.121	0.116	0.116	0.000	0.071	0.053
	Recall	0.373	0.124	0.106	0.153	0.000	0.060	0.060
Blog-5	F-measure	0.132	0.157	0.157	0.153	—	0.074	0.074
	Precision	0.076	0.103	0.104	0.101	0.000	0.083	0.074
	Recall	0.496	0.330	0.322	0.315	0.000	0.067	0.074
Blog-6	F-measure	0.128	0.128	0.130	0.121	—	0.066	0.050
	Precision	0.076	0.108	0.109	0.105	0.000	0.073	0.047
	Recall	0.396	0.162	0.176	0.155	0.000	0.060	0.053

Compared with the Wikipedia datasets' experiment, the Blogcatalog datasets' experiment achieved lower accuracy in general, which might be caused by the larger number of mutual role labels.

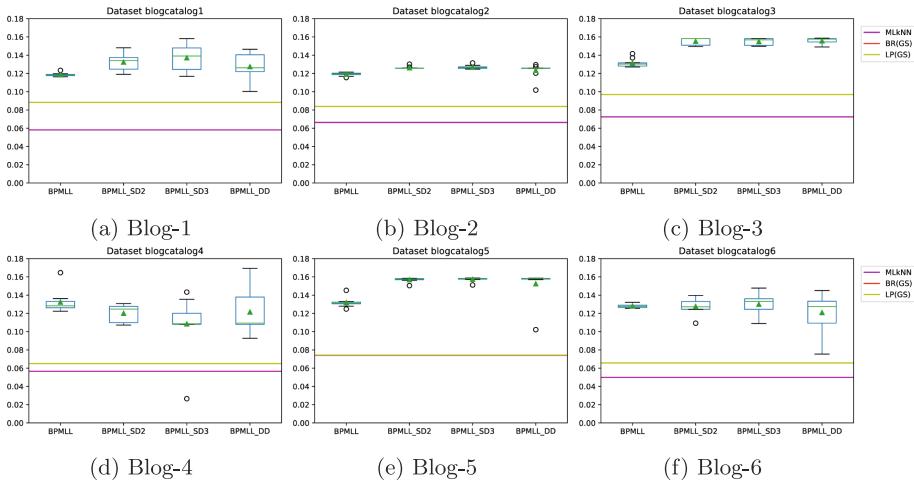


Fig. 3. Boxplots for F-measure results of Blogcatalog network experiment

6 Conclusion

In this study, we proposed a method for multiple-role discovery in complex networks. We also proposed a method for creating training data by community extraction for role discovery in real-world networks. The proposed method achieves higher accuracy in real-world networks such as the Wikipedia and Blogcatalog networks, implying that it can be applied to a variety of real-world networks.

Future work from this research includes optimization of the proposed method's variations, improvement of the efficiency of hyperparameter tuning, improvement of the accuracy of multiple role discovery for a large number of labels, and application of methods that can efficiently obtain distributed representations.

References

1. Rossi, R.A., Ahmed, N.K.: Role discovery in networks. *IEEE Trans. Knowl. Data Eng.* **27**(4), 1112–1131 (2014)
2. Wang, J., Xinghuo, Yu., Stone, L.: Effective augmentation of complex networks. *Sci. Rep.* **6**(1), 1–9 (2016)
3. Kikuta, S., et al.: Framework for role discovery using transfer learning. *Appl. Netw. Sci.* **5**(1), 1–19 (2020). <https://doi.org/10.1007/s41109-020-00281-3>
4. Zhang, J., Dong, Y., Wang, Y., Tang, J., Ding, M.: ProNE: fast and scalable network representation learning. In: *IJCAI*, vol. 19, pp. 4278–4284 (2019)
5. Arabie, P., Boorman, S.A., Levitt, P.R.: Constructing blockmodels: how and why. *J. Math. Psychol.* **17**(1), 21–63 (1978)
6. Holland, P.W., Laskey, K.B., Leinhardt, S.: Stochastic blockmodels: first steps. *Soc. Netw.* **5**(2), 109–137 (1983)

7. Nowicki, K., Snijders, T.A.B.: Estimation and prediction for stochastic blockstructures. *J. Am. Statist. Assoc.* **96**, 9 (2001)
8. Batagelj, V., Mrvar, A., Ferligoj, A., Doreian, P.: Generalized blockmodeling with pajek. *Metodoloski zvezki* **1**(2), 455–467 (2004)
9. Airoldi, E.M., Blei, D.M., Fienberg, S.E., Xing, E.P.: Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.* **9**, 1981–2014 (2008)
10. Burt, R.S.: Positions in networks. *Soc. Forces* **55**(1), 93–122 (1976)
11. Brandes, U., Lerner, J.: Structural similarity: spectral methods for relaxed block-modeling. *J. Classif.* **27**(3), 279–306 (2010)
12. Herrera, F., Charte, F., Rivera, A.J., del Jesus, M.J.: Multilabel classification. In: *Multilabel Classification*. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41111-8_2
13. Zhang, M.-L., Zhou, Z.-H.: ML-KNN: a lazy learning approach to multi-label learning. *Pattern Recognit.* **40**(7), 2038–2048 (2007)
14. Ganin, Y., et al.: Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **17**(1), 1–35 (2016)
15. Zhang, M.-L., Zhou, Z.-H.: Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Trans. Knowl. Data Eng.* **18**(10), 1338–1351 (2006)
16. Grodzicki, R., Mańdziuk, J., Wang, L.: Improved multilabel classification with neural networks. In: Rudolph, G., Jansen, T., Beume, N., Lucas, S., Poloni, C. (eds.) *PPSN 2008. LNCS*, vol. 5199, pp. 409–416. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87700-4_41
17. Traag, V.A., Waltman, L., Van Eck, N.J.: From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**(1), 1–12 (2019)
18. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech. Theor. Exp.* **2008**(10), P10008 (2008)



Graph Summarization with Latent Variable Probabilistic Models

Shintaro Fukushima^{1,2(✉)}, Ryoga Kanai¹, and Kenji Yamanishi¹

¹ The University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo, Japan
s.fukushima@mail.toyota.co.jp, ryoga_kanai@mist.i.u-tokyo.ac.jp,
yamanishi@g.ecc.u-tokyo.ac.jp

² Toyota Motor Corporation, Otemachi 1-6-1, Chiyoda-ku, Tokyo, Japan

Abstract. This study addresses the issue of summarizing a static graph, known as graph summarization, effectively and efficiently. The resulting compact graph is referred to as a summary graph. Based on the minimum description length principle (MDL), we propose a novel graph summarization algorithm called the graph summarization with latent variable probabilistic models (GSL) for a static graph. MDL asserts that the best statistical decision strategy is the one that best compresses the data. The key idea of GSL is encoding the original and summary graphs simultaneously using latent variable probabilistic models with two-part coding, that is, first encoding a summary graph, then encoding the original graph given the summary graph. When encoding these graphs, we can use various latent variable probabilistic models. Therefore, we can encode a more complex graph structure than the conventional graph summarization algorithms. We demonstrate the effectiveness of GSL on both synthetic and real datasets.

Keywords: Graph summarization · Machine learning and data mining · Information theory · Latent variable probabilistic model · Minimum description length principle · Normalized maximum likelihood code-length

1 Introduction

1.1 Motivation

Recently, an increasing number of graphs have been collected in the fields of science [9, 10], social science [11, 14], and engineering. Accordingly, the sizes of such graphs have increased. However, many algorithms and methods used for graph analysis cannot be applied to such larger graphs owing to high computational costs. Therefore, it has become even more important to summarize a graph in an effective and efficient manner.

A promising approach to realizing such a summary is *graph summarization* (e.g., [12]). Graph summarization aims to summarize important attributes or parts of a static or dynamic graph. The compact graph obtained using a

graph summarization is called a *summary graph*. The primary concern of graph summarization is how compactly and quickly it can summarize a given graph.

In this study, a novel graph summarization algorithm called the graph summarization with latent variable probabilistic models (GSL) is proposed. GSL aims to obtain a summary graph from an information-theoretic perspective. Using GSL, we can encode more complex graph structures compared to conventional graph summarization algorithms. This includes a broader class of graphs, such as graphs with weighted edges.

1.2 Related Work

Graph summarization has been extensively explored in the areas of data mining and knowledge discovery [6–8, 15, 29, 33]. Please refer to the comprehensive survey in [12] for further details.

Several studies have addressed the issue of graph summarization based on the minimum description length (MDL) principle. MDL asserts that the best statistical decision strategy is the one that best compresses data. Navlakha et al. [15] proposed an algorithm with a bounded error based on MDL. The idea of their algorithm is to merge supernodes (representative nodes) by minimizing the total code-length for encoding supernodes and superedges (edges between supernodes), as well as correcting mistakes. LeFevre and Terzi [8] proposed an algorithm called k-Gs based on MDL, which repeatedly merges nodes to minimize an ℓ_1 reconstruction error. Lee et al. [7] introduced a two-part coding scheme [22] and proposed a sparse graph summarization algorithm called SSumM. SSumM merges nodes and repeatedly drops less important edges to make the graph more sparse. Zhou et al. [33] proposed an algorithm called DPGS for obtaining a degree-preserving summary graph. In addition, Koutra et al. [6] proposed a graph summarization algorithm called VoG, which is a vocabulary-based summarization scheme that encodes a graph with frequent patterns, such as cliques, stars, and bipartite cores. Note that several algorithms using MDL have been proposed in the fields of network science for community detection [4, 16–20], and model selection and link prediction [27].

1.3 Significance and Novelty

The significance and novelty of this study are stated as follows.

1. Novel Framework for Graph Summarization: We propose a novel graph summarization algorithm called GSL based on MDL. As mentioned in the previous studies [6–8, 33], MDL is applicable to graph summarization because with MDL the complex structure of a graph can be encoded, and subsequently, the best summary graph can be selected based on a sound theoretical background.

The key idea of GSL is to encode the original and summary graphs simultaneously by using the decomposed normalized maximum likelihood (DNML) code-length [28, 32]. The use of DNML code-length produces a compression

effect that cannot be achieved simply by using MDL. When DNML code-length is calculated for a graph, the data-generating mechanism of the original graph is modeled from a summary graph with a latent variable probabilistic model. Although several studies have previously addressed the issue of graph summarization with MDL, and some studies have investigated graph summarization of attributed graphs [29], to the best of our knowledge, there have been no studies modeling the distribution of nodes and edges with a latent variable probabilistic model. Previous studies have summarized the number of nodes and edges of a graph with MDL only in terms of “bits.”

By contrast, we can encode a more complex structure of graphs using GSL. Therefore, we summarize many types of graphs with GSL, including graphs with weighted edges, which is beyond the capabilities of most of conventional graph summarization algorithms. Moreover, we demonstrate a sound theoretical background based on DNML code-length [28, 32].

2. **Empirical Demonstration of the Effectiveness of the Proposed Algorithm:** We demonstrate the effectiveness of the proposed algorithm on synthetic and real datasets. We compared the performance of GSL with other graph summarization algorithms, and confirmed that GSL is superior to these algorithms in terms of the trade-off between compression ratio and reconstruction error.

2 Proposed Algorithm

In this section, we propose a novel graph summarization algorithm called graph summarization with latent variable probabilistic models (GSL) for static graphs. The key idea of our algorithm is to minimize the code-lengths required for encoding the original and summary graphs simultaneously with MDL using probabilistic models. The details of MDL are described in Appendix A.

2.1 Problem Setting

Let $G = (V, E)$ be a directed or undirected graph, where V and E denote a set of nodes and a set of edges, respectively. We denote the number of nodes as $|V|$.

First, we define a summary graph of G as follows:

Definition 1 (*Summary graph*). [7] *A summary graph \bar{G} of graph G is defined as $\bar{G} = (\bar{S}, \bar{U}, \omega)$. S is a set of supernodes such that $\cup_{A \in S} A = V$ and $\forall A, B \in S$, $A \cap B = \emptyset$. Note that supernodes refer to distinct and exhaustive subsets of nodes in G . U is a set of superedges, where superedges refer to edges between supernodes. $\omega : U \rightarrow \mathbb{R}^+$ is a weight function for superedges.*

The weight function ω maps a superedge in U to a non-zero number. The choice of ω depends on how to summarize the graph. In [7], the authors defined ω as the total number of edges between supernodes.

Next, we define a summary graph of G when we apply a latent variable probabilistic model as follows:

Definition 2 (*Summary graph with latent variable probabilistic model*). A summary graph \bar{G} of graph G with a latent variable probabilistic model is defined as $\bar{G} = (S, U, \eta, \xi)$, where S and U are the same as in the general definition given in Definition 1, $\eta : U \rightarrow [0, 1]^{|U| \times |U|}$ is a probability function for superedges, and $\xi : S \rightarrow [0, 1]^{|S|}$ is a probability function for supernodes.

The meanings of η and ξ would be clarified in Sect. 2.2.

Our problem setting is as follows. Given a directed or undirected graph G , we find a summary graph \bar{G} that satisfies the following equation:

$$L(G, \bar{G}) = L(G|\bar{G}) + L(\bar{G}) \implies \min . \text{ w.r.t. } \bar{G}, \quad (1)$$

where $L(G, \bar{G})$ is the code-length required for encoding the original and summary graphs simultaneously. In addition, $L(\bar{G})$ is the code-length required for encoding \bar{G} , whereas $L(G|\bar{G})$ is the code-length required for encoding G given \bar{G} . Therefore, we encode the original and summary graphs in two steps. Equation (1) is a specific form of the so-called two-part coding [24] in Eq. (14). See Appendix A for this coding.

2.2 Code-Lengths for Encoding Graphs

We consider how to achieve Eq. (1). To do so, we first consider the two code-lengths separately, and then integrate and minimize them using the two-part coding. First, the code-length required for encoding a summary graph \bar{G} is defined as the code-length for encoding the number of supernodes: $L(\bar{G}) \stackrel{\text{def}}{=} L_{\mathbb{N}}(k)$, where k denotes the number of supernodes in \bar{G} . Here, $L_{\mathbb{N}}(k)$ is the optimal encoding length for a positive integer k [23]: $L_{\mathbb{N}}(k) = 2.865 + \log k + \log \log k + \dots$, where the sum is taken over the positive terms.

Next, we consider how to encode graph G given the summary graph \bar{G} , that is, the code-length $L(G|\bar{G})$ in Eq. (1). Because our study aims at modeling nodes and edges to flexibly represent a graph, we introduce a probabilistic model. Let us denote random variables X and Z . Here, X denotes the connection between nodes, and Z denotes the membership of each node. In addition, $x = (x_{ij})_{i,j=1,\dots,N}$ and $z = (z_i)_{i=1,\dots,N}$ are the realizations of X and Z , respectively. Moreover, x_{ij} denotes a weight between node i and j , and z_i denotes the group that node i belongs to. We assume that the joint probability of X and Z is decomposed as follows:

$$p(X, Z; \theta, k) = p_{X|Z}(X|Z; \eta, k)p_Z(Z; \xi, k), \quad (2)$$

where $\theta = (\eta, \xi)$ is a parameter, η is a connection parameter between supernodes, ξ is a membership parameter for supernodes, and k is the number of supernodes, as before. Note that η and ξ were used in the definition of summary graph with a latent variable probabilistic model in Definition 2.

Equation (2) includes many important classes of probability distributions, in particular, classes of latent variable probabilistic models [28, 32]. As an example, for the stochastic block model (SBM) [26], Eq. (2) is expressed as follows:

$$z_i \sim \text{Multinomial}(z_i; \xi, k), \quad x_{ij} \sim \text{Bernoulli}(x_{ij}; \eta, k) \quad (i, j = 1, \dots, |V|), \quad (3)$$

where Multinomial and Bernoulli denote the multinomial and Bernoulli distributions, respectively. Note that $p_{X|Z} = \text{Bernoulli}$ and $p_Z = \text{Multinomial}$ in Eq. (3).

We then consider how to encode nodes and edges in the original graph G given a summary graph \bar{G} if each node and edge is drawn from Eq. (2). We introduce the decomposed normalized maximum likelihood (DNML) code-length [28, 32] to encode the graph efficiently with a sound theoretical guarantee. The use of DNML produces a compression effect that cannot be achieved simply by using MDL. DNML code-length is defined as follows:

$$L_{\text{DNML}}(x, z; k) \stackrel{\text{def}}{=} L_{\text{NML}}(x|z; k) + L_{\text{NML}}(z; k) \quad (4)$$

Here,

$$L_{\text{NML}}(x|z; k) \stackrel{\text{def}}{=} -\log p_{X|Z}(x|z; \hat{\eta}(x, z), k) + \log \sum_{x'} p_{X|Z}(x'|z; \hat{\eta}(x', z), k) \quad (5)$$

$$L_{\text{NML}}(z; k) \stackrel{\text{def}}{=} -\log p_Z(z; \hat{\xi}(z), k) + \log \sum_{z'} p_Z(z'; \hat{\xi}(z'), k), \quad (6)$$

where $\hat{\eta}$ and $\hat{\xi}$ are the maximum likelihood estimators of η and ξ , respectively. The NML code-length is the optimal code-length that achieves Shtarkov's minimax regret [25]. It is frequently used in model selection [28, 30, 32], change detection [2], and many other important tasks in data mining and knowledge discovery. The second terms in Eqs. (5) and (6) are the normalization terms, called the parametric complexity [3]. See Appendix A for details. Then, we define the total code-length required for encoding all nodes and edges in G given a summary graph \bar{G} with DNML code-length:

$$L(G|\bar{G}) \stackrel{\text{def}}{=} L_{\text{DNML}}(x, z; k) = L_{\text{NML}}(x|z; k) + L_{\text{NML}}(z; k).$$

Note that DNML code-length was originally designed for model selection of latent variable probabilistic models [28, 32] and later applied to change detection from data streams [2]. For example, let us consider SBM [26]. The code-length is calculated as [28, 32]:

$$\begin{aligned} L(G|\bar{G}) &= \sum_{k_1, k_2} (n_{k_1 k_2} \log n_{k_1 k_2} - n_{k_1 k_2}^+ \log n_{k_1 k_2}^+ - n_{k_1 k_2}^- \log n_{k_1 k_2}^-) \\ &\quad + \sum_{k_1, k_2} \log C(n_{k_1 k_2}, 2) + \sum_k n_k (\log n - \log n_k) + \log C(n, k), \end{aligned}$$

where $n_{k_1 k_2}^+$ and $n_{k_1 k_2}^-$ are the numbers of links and non-links in groups (k_1, k_2) , and n_k is the number of nodes belonging to group k . In addition, n_{k_1, k_2} is defined as $n_{k_1, k_2} = n_{k_1 k_2}^+ + n_{k_1 k_2}^-$, that is, the total number of links in groups (k_1, k_2) . Moreover, $\log C(n, k)$ is the normalization term for data of size n following the multinomial distribution with k elements, and is computable with $O(n+k)$ [5].

Another example is the weighted stochastic block model for the Poisson distribution:

$$z_i \sim \text{Multinomial}(z_i; \xi, k), \quad x_{ij} \sim \text{Poisson}(x_{ij}; \eta, k) \quad (i, j = 1, \dots, |V|),$$

where Poisson denotes the Poisson distribution. We have only to consider the following term for encoding the data given a summary graph compared to SBM in Eq. (3):

$$-\log p_{X|Z}(x|z; \hat{\eta}(x, z), k) + \log \sum_{x'} p_{X|Z}(x'|z; \hat{\eta}(x', z), k). \quad (7)$$

The first term in Eq. (7) is easily calculated using the maximum likelihood estimator of the Poisson distribution. By contrast, it is known that the second term in Eq. (7) is approximated using the Rissanen's approximate formula [24]. Please refer to [24, 31] for further details.

Consequently, we can flexibly model the nodes and edges with probabilistic models within the framework of graph summarization, which is one of the primary advantages of GSL over previous studies.

2.3 Algorithm of GSL

In summary, the problem of estimating a summary graph \bar{G} for a given graph G is as follows:

$$\begin{aligned} \min_{\bar{G}} L(G, \bar{G}) &\stackrel{\text{def}}{=} \min_k \{L(G|\bar{G}) + L(\bar{G})\} \\ &\stackrel{\text{def}}{=} \min_k \left(L_{\mathbb{N}}(k) - \log p_{X|Z}(x|z; \hat{\eta}(x, z), k) + \log \sum_{x'} p_{X|Z}(x'|z; \hat{\eta}(x', z), k) \right. \\ &\quad \left. - \log p_Z(z; \hat{\xi}(z), k) + \log \sum_{z'} p_Z(z'; \hat{\xi}(z'), k) \right). \end{aligned} \quad (8)$$

Hence, we estimate the number of supernodes k , and the parameters of the distribution $\theta = (\eta, \xi)$. Note that GSL does not search for supernodes or superedges directly. The two-part coding approach of GSL is similar to several previous studies on graph summarization of a static graph [7, 33]; however, these studies focus on encoding the number of nodes and edges. By contrast, our study uses a probability distribution of groups of nodes and one of edges between the nodes to flexibly model them. In practice, we estimate the number of supernodes k and associated parameters $\theta = (\eta, \xi)$ for SBM (e.g., [13, 19]), for example. The algorithm is summarized in Algorithm 1.

Note that the summary graph \bar{G} is sometimes not sparse due to the misspecification of SBM. For a more sparse summary graph, we drop superedges in \bar{G} in a similar manner that is adopted in [7]. Please refer to Sect. 3 for details.

3 Experiments

In this section, to demonstrate the effectiveness of GSL, we describe experiments conducted on synthetic and real datasets.

Algorithm 1. Graph summarization with probabilistic latent variable model algorithm (GSL)

Input: \mathcal{K} : set of supernode numbers.

Output: \hat{k} : the optimal number of supernodes. $\hat{\eta}$: the estimated parameter of probability distribution between nodes. $\hat{\xi}$: the estimated parameter of probability distribution of nodes.

1: **for** $k \in \mathcal{K}$ **do**

2: Estimate blocks \hat{z} .

3: Calculate code-lengths according to Eq. (4).

4: **end for**

5: Choose the best k that satisfies Eq. (8) and set $\hat{k} = k$. Thus, obtain the summary graph $\bar{G} = (S, U, \hat{\eta}, \hat{\xi})$.

3.1 Synthetic Dataset

Dataset. We generated static undirected graphs from SBM [26] according to Eq. (3). We set the number of nodes to $|V| = 1000$ and the number of blocks to $K = 3$. We also set the parameters of SBM as follows:

$$\eta = \begin{pmatrix} 0.7 & 0.2 & 0.3 \\ 0.2 & 0.7 & 0.4 \\ 0.3 & 0.4 & 0.7 \end{pmatrix}, \quad \xi = \begin{pmatrix} 0.2 \\ 0.5 \\ 0.3 \end{pmatrix}. \quad (9)$$

Methods for Comparison. We compared GSL with the following two algorithms.

- SSumM [7]: SSumM is a two-part coding summarization algorithm and requires a size of k for a summary graph as an input. We changed k from 0.05 to 0.95 by increments of 0.05. We used the open-sourced code at <https://github.com/KyuhanLee/SSumM>.
- k-GS [8]: As an iterative merging scheme, k-GS takes a computational cost of $O(|V|^3)$. We used the SamplePair method with $c = 1$, as in [7], which is one of the three methods proposed in [8]. The target summary size changed from 10% to 50% by increments of 10%.

Evaluation Metrics. We evaluated the performance of GSL against those of the other methods described above, using the reconstruction error and size of the summary graph. There is a trade-off between them. The reconstruction error is defined as the following ℓ_1 -error of the adjacency matrices between the original and summary graphs [1, 7, 8, 21]:

$$e \stackrel{\text{def}}{=} \sum_{i,j=1}^{|V|} \frac{|A_{i,j} - \bar{A}_{i,j}|}{|V|(|V| - 1)}, \quad (10)$$

where $A_{i,j}$ is the edge weight between node i and j in the original graph G , and $\bar{A}_{i,j}$ is the one in the summary graph. $|V|$ is the number of nodes in G . Note that the reconstruction error with GSL is mainly due to the modeling with SBM.

The size of summary graph $\bar{G} = (\bar{S}, \bar{U}, \omega)$ is defined as follows [7]:

$$\text{Size}(\bar{G}) \stackrel{\text{def}}{=} |\bar{U}|(2\log_2 |\bar{S}| + \log_2 \omega_{\max}) + |\bar{V}| \log_2 |\bar{S}|, \quad (11)$$

where $|\bar{U}|$ denotes the number of superedges in \bar{G} , and ω_{\max} denotes the maximum superedge weight. The first term in Eq. (11) represents the code-length required for encoding each edge with two supernodes and an edge weight, i.e., $2\log_2 S$ and $\log_2 \omega_{\max}$, respectively. The second term in Eq. (11) represents the code-length required for encoding each node: The code-length is $\log_2 |S|$ because it is necessary to distinguish $|S|$ different supernodes.

We used the following relative size of a summary graph as a ratio of the sizes of the summary graph and original graph $G = (V, E)$:

$$\text{RelSize}(\bar{G}) \stackrel{\text{def}}{=} \frac{\text{Size}(\bar{G})}{\text{Size}(G)}, \quad (12)$$

where the denominator of Eq. (12) denotes the size of G , i.e., the code-length required for encoding each edge with two nodes, that is, $\text{Size}(G) = 2|E| \log_2 |V|$, where $|V|$ and $|E|$ denote the number of nodes and edges in G , respectively.

Note that the summary graph \bar{G} is sometimes not sparse. For a more sparse summary graph, we drop each superedge in \bar{G} in a similar manner that is adopted in [7]. We find $\kappa = \lceil (\text{Size}(\bar{G}) - r)/(2\log_2 |\bar{S}| + \log_2 \omega_{\max}) \rceil$ -th smallest increase in the reconstruction error and let it be Δ_κ . Here, Size is the graph size defined in Eq. (11), and r denotes the target size of the summary graph. We finally drop each superedge in the summary graph \bar{G} if the reconstruction error is smaller than or equal to Δ_κ .

In the following experiments, we generated graphs drawn from SBM using the parameter specified in Eq. (9) five times and took the averages for both metrics. We set the number of the blocks of SBM to $\lceil r|V| \rceil$, where r changed from 0.05 to 0.95 by increments of 0.05. We also used r as the target size of summary graph in calculating Δ_κ , which is described in the previous paragraph.

Results. Figure 1 shows the trade-off curve for the relative size of a summary graph and reconstruction error. We can observe that when the relative size of a summary graph is over 0.4, GSL has lower reconstruction errors compared to SSumM. However, when its size is less than 0.4, GSL is comparable to SSumM. The ability of GSL to provide compact summary graphs decreased as the relative size decreased because the number of supernodes of the summary graph obtained with GSL decreased and the effect of the sparsification of superedges did not work well. In this case, k-Gs performed far worse than GSL and SSumM.

3.2 Real Dataset

We used the ego-Facebook dataset [14], which is available at <http://snap.stanford.edu/data/ego-Facebook.html>. This dataset consists of undirected

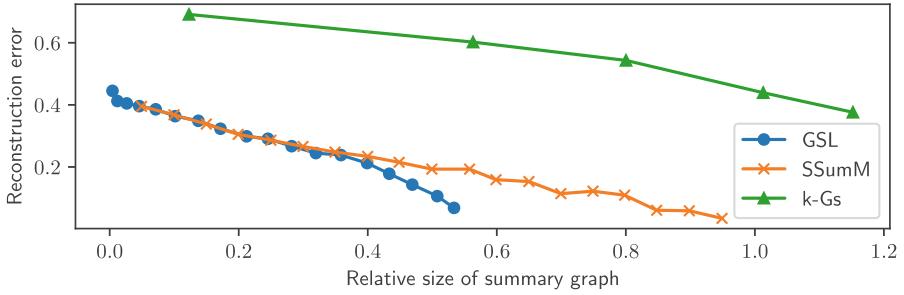


Fig. 1. Evaluation results for GSL, SSumM, and k-GS for the synthetic dataset drawn from SBM. The horizontal and vertical axes indicate the relative size of the summary graph and the reconstruction error in Eqs. (12) and (10), respectively.

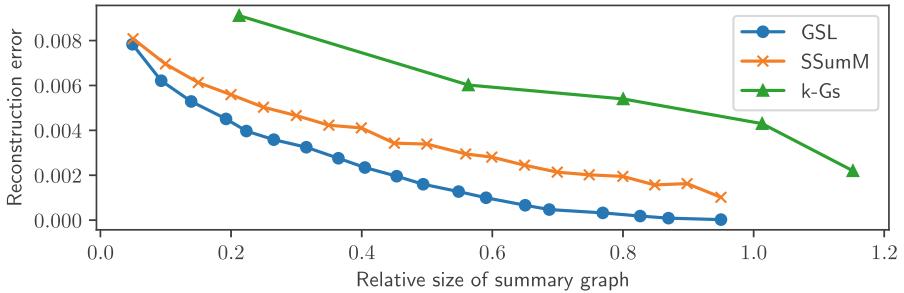


Fig. 2. Evaluation results for GSL, SSumM, and k-GS for the ego-Facebook dataset.

friends lists from Facebook. There are 4039 nodes and 88,234 edges. We used the same evaluation metrics as applied in the experiment on the synthetic dataset, i.e., the reconstruction error in Eq. (10) and the relative size of the summary graph in Eq. (11).

Figure 2 shows the trade-off curve for the relative size of a summary graph and the reconstruction error. We observed that GSL has lower reconstruction errors compared to the other algorithms within all ranges of the relative size of the summary graph. The ego-Facebook dataset is composed of various groups of friends. Therefore, we infer that GSL was still superior in representing such a variety of graph structures, even within the ranges of the higher relative sizes of the summary graph.

4 Conclusion

In this study, using MDL, a novel graph summarization algorithm called GSL was proposed. The key idea of the proposed algorithm is to encode the original and summary graphs simultaneously with two-part code-lengths. By plugging a latent variable probabilistic model into the code-lengths, our algorithm flexibly

models the distribution of nodes and the connections between them, which leads to a more expressive summary graph. We demonstrated the effectiveness of the proposed algorithm on both synthetic and real datasets. Future studies will include an extension of the proposed algorithm to graph streams.

Acknowledgements. This work was partially supported by JST KAKENHI 191400000190 and JST-AIP JPMJCR19U4.

A MDL Principle

Two-Part Coding. Let Y be a random variable and y be a realization. Assume that we observe a set of data $y^n = y_1, \dots, y_n \in \mathcal{Y}^n$, where $y_i \in \mathbb{R}^d$ ($i = 1, \dots, n$), and \mathcal{Y} denotes the data domain. The fundamental idea of the MDL principle is to select the best model that minimizes the total code-lengths of the model and data. Let us consider the following parametric model families:

$$\mathcal{F} = \{f(Y^n; \theta, M \in \mathcal{M}) : \theta \in \Theta\}, \quad (13)$$

where \mathcal{M} is a model space and M is a model. Here, in the case of SBM, a model refers to a family of models with the fixed number of groups (clusters), and \mathcal{M} is a set of the family of models with various numbers of groups (clusters). Let us consider how to encode data y^n and model M , simultaneously. When we use a prefix code-length L , the total code-length required to encode data y^n and M is decomposed into a sum of the code-length of the data given the model and that of the model itself as

$$L(y^n : M) = L(y^n | M) + L(M), \quad (14)$$

where $L(y^n : M)$, $L(y^n | M)$, and $L(M)$ denote the total code-length required to encode y^n and M , the code-length to encode y^n for the given M , and M , respectively. The prefix code-length refers to the code-length encoded with the prefix coding that satisfies the following Kraft's inequalities [24]: $\sum_{y \in \mathcal{Y}} 2^{-L(y|M)} \leq 1$ and $\sum_{M \in \mathcal{M}} 2^{-L(M)} \leq 1$. Then, the MDL criterion asserts that the total code-length $L(y^n : M)$ should be minimized with respect to M :

$$L(y^n : M) = L(y^n | M) + L(M) \implies \text{min. w.r.t. } M.$$

Normalized Maximum Likelihood (NML) Code-Length. We consider how to achieve the shortest code-length for a family of distributions in Eq. (13). Here, we introduce the normalized maximum likelihood (NML) code-length. The NML code-length is the optimal code-length that achieves Shtarkov's minimax regret [25]. We consider a model class in Eq. (13). The NML code-length achieves the following Shtarkov's minimax regret:

$$\min_g \max_{y^n \in \mathcal{Y}^n} \left\{ -\log g(y^n) - \min_\theta (-\log f(y^n; \theta, M)) \right\},$$

where g is called the normalized maximum likelihood (NML) distribution, a distribution that achieves the minimum regret. The NML distribution f_{NML} is described as follows: $f_{\text{NML}}(y^n) = \frac{f(y^n; \hat{\theta}(y^n), M)}{\int f(Y^n; \hat{\theta}(Y^n), M) dY^n}$, where $\hat{\theta}(y^n)$ is the following maximum likelihood estimator: $\hat{\theta}(y^n) \stackrel{\text{def}}{=} \operatorname{argmax}_{\theta} f(y^n; \theta, M)$. It is possible to encode y^n with the following code-length using the NML distribution:

$$\begin{aligned} L_{\text{NML}}(y^n; M) &\stackrel{\text{def}}{=} -\log \frac{f(y^n; \hat{\theta}(y^n), M)}{\int f(Y^n; \hat{\theta}(Y^n), M) dY^n} \\ &= -\log f(y^n; \hat{\theta}(y^n), M) + \log \int_{Y^n} f(Y^n; \hat{\theta}(Y^n), M) dY^n. \end{aligned} \quad (15)$$

The code-length $L_{\text{NML}}(y^n; M)$ in Eq. (15) is called the NML code-length and the last term in the right-hand side is called the parametric complexity to \mathcal{F} in Eq. (13) with data length n .

References

1. Beg, M.A., Ahmad, M., Zaman, A., Khan, I.: Scalable approximation algorithm for graph summarization. In: Proceedings of the 22nd Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2018, pp. 502–514 (2018)
2. Fukushima, S., Yamanishi, K.: Hierarchical change detection in latent variable models. In: Proceedings of 2020 IEEE International Conference on Data Mining, ICDM2020, pp. 1128–1134 (2020)
3. Grünwald, P.: The Minimum Description Length Principle. MIT Press (2007)
4. Hric, D., Peixoto, T.P., Fortunato, S.: Network structure, metadata, and the prediction of missing nodes and annotations. *Phys. Rev. X* **6**(3), 031038 (2016)
5. Kontkanen, P., Myllymäki, P.: A linear-time algorithm for computing the multinomial stochastic complexity. *Inf. Process. Lett.* **103**(6), 227–233 (2007)
6. Koutra, D., Kang, U., Vreeken, J., Faloutsos, C.: VOG: summarizing and understanding large graphs. In: Proceedings of the 2014 SIAM International Conference on Data Mining, SDM 2014, pp. 91–99 (2014)
7. Lee, K., Jo, H., Ko, J., Lim, S., Shin, K.: SSUMM: sparse summarization of massive graphs. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2020, pp. 144–154 (2020)
8. LeFevre, K., Terzi, E.: GraSS: graph structure summarization. In: Proceedings of the 2010 SIAM International Conference on Data Mining, SDM 2010, pp. 454–465 (2010)
9. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graphs over time: densification laws, shrinking diameters and possible explanations. In: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2005, pp. 177–187 (2005)
10. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data* **1**(1), 2-es (2007)
11. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters. *Internet Math.* **6**(1), 29–123 (2009)

12. Liu, Y., Safavi, T., Dighe, A., Koutra, D.: Graph summarization methods and applications: a survey. *ACM Comput. Surv.* **51**(3), 62:1-62:34 (2018)
13. Mariadassou, M., Robin, S., Vacher, C.: Uncovering latent structure in valued graphs: a variational approach. *Ann. Appl. Stat.* **4**(2), 715–742 (2010)
14. McAuley, J., Leskovec, J.: Learning to discover social circles in ego networks. In: *Proceedings of the 25th Advances in Neural Information Processing Systems, NIPS 2012*, pp. 539–547 (2012)
15. Navlakha, S., Rastogi, R., Shrivastava, N.: Graph summarization with bounded error. In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD 2008*, pp. 419–432 (2008)
16. Peixoto, T.P.: Entropy of stochastic blockmodel ensembles. *Phys. Rev. E* **85**(5), 056122 (2012)
17. Peixoto, T.P.: Parsimonious module inference in large networks. *Phys. Rev. Lett.* **110**, 148701 (2013)
18. Peixoto, T.P.: Efficient Monte Carlo and greedy heuristic for the inference of stochastic block models. *Phys. Rev. E* **89**, 012804 (2014)
19. Peixoto, T.P.: Hierarchical block structures and high-resolution model selection in large networks. *Phys. Rev. X* **4**, 011047 (2014)
20. Peixoto, T.P.: Model selection and hypothesis testing for large-scale network models with overlapping groups. *Phys. Rev. X* **5**, 011033 (2015)
21. Riondato, M., García-Soriano, D., Bonchi, F.: Graph summarization with quality guarantees. *Data Min. Knowl. Disc.* **31**(2), 314–349 (2016). <https://doi.org/10.1007/s10618-016-0468-8>
22. Rissanen, J.: Modeling by shortest data description. *Automatica* **14**, 465–471 (1978)
23. Rissanen, J.: A universal prior for integers and estimation by minimum description length. *Ann. Stat.* **11**(2), 416–431 (1983)
24. Rissanen, J.: Optimal Estimation of Parameters (2012)
25. Shtarkov, Y.M.: Universal sequential coding of single messages. *Probl. Inf. Transm.* **23**(3), 3–17 (1987)
26. Snijders, T.A.B., Nowicki, K.: Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *J. Classif.* **14**(1), 75–100 (1997)
27. Catalá, T.V., Peixoto, T.P., Pardo, M.S., Guimerá, R.: Consistencies and inconsistencies between model selection and link prediction in networks. *Phys. Rev. E* **97**, 062316 (2018)
28. Wu, T., Sugawara, S., Yamanishi, K.: Decomposed normalized maximum likelihood codelength criterion for selecting hierarchical latent variable models. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2017*, pp. 1165–1174 (2017)
29. Wu, Y., Zhong, Z., Xiong, W., Jing, N.: Graph summarization for attributed graphs. In: *Proceedings of 2014 International Conference on Information Science, Electronics and Electrical Engineering, ISEE 2014*, vol. 1, pp. 503–507 (2014)
30. Yamanishi, K., Fukushima, S.: Model change detection with the MDL principle. *IEEE Trans. Inf. Theor.* **9**(64), 6115–6126 (2018)
31. Yamanishi, K., Miyaguchi, K.: Detecting gradual changes from data stream using MDL-change statistics. In: *Proceedings of 2016 IEEE International Conference on BigData, BigData 2016*, pp. 156–163 (2016)

32. Yamanishi, K., Wu, T., Sugawara, S., Okada, M.: The decomposed normalized maximum likelihood code-length criterion for selecting hierarchical latent variable models. *Data Min. Knowl. Disc.* **33**(4), 1017–1058 (2019). <https://doi.org/10.1007/s10618-019-00624-4>
33. Zhou, H., Liu, S., Lee, K., Shin, K., Shen, H., Cheng, X.: DPGS: degree-preserving graph summarization. In: Proceedings of the 2021 SIAM International Conference on Data Mining, SDM 2021, pp. 280–288 (2021)



Image Keypoint Matching Using Graph Neural Networks

Nancy Xu^{1(✉)}, Giannis Nikolentzos², Michalis Vazirgiannis^{1,3},
and Henrik Boström¹

¹ KTH Royal Institute of Technology, Stockholm, Sweden
{nancyx,mvaz,bostromh}@kth.se

² Athens University of Economics and Business, Athens, Greece
nikolentzos@auer.gr

³ École Polytechnique, Palaiseau, France

Abstract. Image matching is a key component of many tasks in computer vision and its main objective is to find correspondences between features extracted from different natural images. When images are represented as graphs, image matching boils down to the problem of graph matching which has been studied intensively in the past. In recent years, graph neural networks have shown great potential in the graph matching task, and have also been applied to image matching. In this paper, we propose a graph neural network for the problem of image matching. The proposed method first generates initial soft correspondences between keypoints using localized node embeddings and then iteratively refines the initial correspondences using a series of graph neural network layers. We evaluate our method on natural image datasets with keypoint annotations and show that, in comparison to a state-of-the-art model, our method speeds up inference times without sacrificing prediction accuracy.

Keywords: Keypoint matching · Graph neural networks · Graph matching

1 Introduction

Image matching, the task of finding correspondences between the key features extracted from one image and those extracted from another image of the same object, is a fundamental task in computer vision. Indeed, image matching lies at the heart of several applications in computer vision including 3D structure recovery, simultaneous localization and mapping (SLAM), and change detection, just to name a few. Over the past decades, a large amount of research was devoted to image matching and led to the development of a diverse set of approaches [23].

The pipeline that is typically followed by most image matching algorithms consists of the extraction of a set of key features (or keypoints) from each image and the generation of descriptors for those features. The above steps are followed

by some matching approach which is responsible for establishing correspondences between the keypoints of the two images. Recently, the last part of the pipeline, i.e., the task of keypoint matching in natural images, has been formulated as a graph matching problem and has been addressed using graph neural network architectures [9, 25, 28]. Images are represented as graphs where nodes correspond to keypoints and edges capture proximity or other types of relations between keypoints. In fact, the problem of graph matching has been studied a lot in the past, but mainly from an algorithmic viewpoint [5]. Unfortunately, most derived formulations correspond to NP-hard problems, and therefore, are not useful for most real-world settings. With the advent of deep learning and the success of graph neural networks, the problem of learning how to compute correspondences of nodes (or of other structures) between two graphs attracted increasing attention [1, 15, 26, 28]. Though not exact, approaches that utilize graph neural networks to find correspondences between nodes are very efficient, while they also learn representations that are best suited for the task at hand.

In this paper, we capitalize on recent advances in graph representation learning, and we propose a graph neural network for establishing meaningful structural correspondences of keypoints between two natural images. More specifically, we take as starting point the Deep Graph Matching Consensus (DGMC) method proposed in [9]. DGMC operates in two stages, a local feature matching stage which generates an initial soft correspondence matrix, and a neighborhood consensus stage which iteratively refines the soft correspondence matrix by repeatedly applying a single graph neural network to the scores. The final correspondence matrix is a combination of the initial soft matrix and the final result of the iterations.

We develop a new approach for obtaining and combining soft correspondence matrices that is more intuitive and more attractive in terms of running time. The proposed method first generates initial soft correspondences between keypoints using localized node embeddings, and then iteratively refines the initial correspondences using a series of graph neural network layers. The final correspondence matrix is formed of all the matrices produced during the consensus stage. This change increases the complexity of the neighborhood consensus stage which reduces the number of iterations needed, thus decreasing inference time. Experiments on standard keypoint matching datasets demonstrate that in comparison to the DGMC, the proposed method is in most cases more efficient, while in terms of performance, it matches or outperforms the baseline model.

The rest of this paper is organized as follows. Section 2 provides an overview of the related work. Section 3 provides a detailed description of the proposed model. Section 4 evaluates the proposed model in the task of keypoint matching in natural images. Finally, Sect. 5 concludes.

2 Related Work

Traditional methods for identifying correspondences between images typically deal with instance matching tasks where correspondences are found between

two images of the same scene or object. These methods use representations such as image pyramids which are formed by repeatedly filtering and subsampling a given image. Image features are computed densely over image pyramids and can be sampled densely [7] or used to identify keypoints [22]. Notably, SIFT-Flow combines the image pyramid with optical flow methods for instance matching [19]. Unfortunately, most traditional methods depend on handcrafted features which must be manually defined.

In comparison, deep learning methods automatically learn important image features and are widely used in computer vision. For instance, Long *et al.* applied in [21] a straightforward convolutional neural network to image matching and achieved similar and in some cases, superior results to those of SIFT-Flow. The introduction of datasets specialized for image matching such as Proposal Flow (PASCAL-PF) [14], WILLOW-ObjectClass [3], and the PASCAL-VOC Berkeley annotations [2], greatly improved the ease of developing new deep learning methods. Examples include FCSS, which uses a convolutional self-similarity layer to encode local self-similarity patterns [16] and SFNet which outputs a semantic flow field using a CNN [18].

Finding correspondences between keypoints in images can also be seen as a graph matching problem. Graph matching can be formulated as a quadratic assignment problem, which turns out to be a notoriously difficult problem [20]. A more computationally tractable formulation is to treat the problem as an optimal transport problem. Xu *et al.* demonstrated in [27] the success of using this formulation along with the Sinkhorn algorithm for graph matching [17]. Graph neural networks can effectively and automatically generate node or graph representations using message passing schemes [13]. Several recent methods use a combination of graph neural networks and the Sinkhorn algorithm for graph matching [9, 25, 26, 28]. By using a graph neural network to generate similarity scores followed by the application of the Sinkhorn normalization, we can build an end-to-end trainable framework for semantic matching between keypoints extracted from images.

3 Graph Neural Networks for Image Keypoint Matching

Before presenting the proposed method, we begin by introducing some key notation for graphs which will be used later. Let $G = (V, E)$ be a graph where V denotes the set of nodes and E the set of edges. We will denote by n the number of vertices and by m the number of edges. Let also $\mathbf{A} \in \mathbb{R}^{n \times n}$ denote the adjacency matrix of G . Following previous work [9], we represent images as graphs where nodes correspond to keypoints and edges are obtained via the Delaunay triangulation of the keypoints. Furthermore, every node in the graph is associated with a feature vector (i.e., the input features of keypoints), and we use $\mathbf{X} \in \mathbb{R}^{n \times d}$ to denote those features where d is the feature dimensionality. The edges of the graph can also be potentially associated with a feature vector (e.g., distance of two endpoints), and we use $\mathbf{E} \in \mathbb{R}^{n \times n \times d}$ to denote those features.

Given a source and a target image, we first map these images to graphs G_s and G_t , respectively. Let n_s and n_t denote the number of nodes of the two graphs.

Without loss of generality, we assume that $n_s \leq n_t$. The goal of the method is to establish correspondences between G_s and G_t and match two pairs of keypoints.

Similar to DGMIC [9], our method consists of a local feature matching stage followed by a neighborhood consensus stage, which are both presented in the following subsections. The local feature matching stage finds initial correspondences between images based on the similarity of their node embeddings and is implemented as a graph neural network layer $\text{GNN}^{(0)}$. The neighborhood consensus stage then refines these initial correspondences which we implement as a series of graph neural network layers $\text{GNN}^{(1)}, \dots, \text{GNN}^{(K)}$.

3.1 Local Feature Matching

Starting with G_s and G_t , the representations of the nodes are updated by feeding both graphs into a graph neural network, i.e., $\mathbf{H} = \text{GNN}^{(0)}(\mathbf{A}, \mathbf{X}, \mathbf{E})$. Since we use a graph neural network to update the features of the keypoints, we obtain localized and permutation equivariant representations. Then, the similarity between keypoints extracted from the two images is determined by the similarity of their node representations \mathbf{H}_s and \mathbf{H}_t . Specifically, we generate matrix $\mathbf{K}^{(0)}$ as follows $\mathbf{K}^{(0)} = \mathbf{H}_s \mathbf{H}_t^\top$, and then, we can obtain initial soft correspondences between the keypoints of the two images via the Sinkhorn algorithm [6]:

$$\mathbf{S}^{(0)} = \text{sinkhorn}(\mathbf{K}^{(0)}) \in [0, 1]^{n_s \times n_t}$$

The Sinkhorn normalization produces a rectangular doubly-stochastic correspondence matrix such that $\sum_{j=1}^{n_t} \mathbf{S}_{i,j}^{(0)} = 1, \forall i \in \{1, \dots, n_s\}$ and $\sum_{i=1}^{n_s} \mathbf{S}_{i,j}^{(0)} \leq 1, \forall j \in \{1, \dots, n_t\}$.

As discussed in [9], the Sinkhorn algorithm is expensive in terms of running time, while it also might lead to vanishing gradients. Furthermore, it may also converge to inconsistent solutions which the next stage of the algorithm might find difficult to refine. Therefore, instead of applying Sinkhorn, the problem is partially relaxed by dropping the following constraint $\sum_{i=1}^{n_s} \mathbf{S}_{i,j}^{(0)} \leq 1, \forall j \in \{1, \dots, n_t\}$, and the initial correspondence matrix is computed by applying row-wise softmax normalization on matrix $\mathbf{K}^{(0)}$ as follows:

$$\mathbf{S}^{(0)} = \text{softmax}(\mathbf{K}^{(0)})$$

3.2 Neighborhood Consensus

The initial correspondences only take local features into account, which makes it more prone to falsely matching nodes which are only superficially similar. The neighborhood consensus step aims to detect these false correspondences and iteratively resolve them. The iteration is done by applying a series of graph neural network layers which we refer to as $\text{GNN}^{(1)}, \dots, \text{GNN}^{(K)}$. In contrast to DGMIC proposed in [9], which uses a shared graph neural network for all iterations, in our case no weight sharing is performed and each graph neural network consists of its own trainable parameters. The idea behind this change is to increase the

complexity of the neighborhood consensus stage while increasing efficiency by reducing the number of iterations needed to refine the correspondence matrices.

At each iteration, a graph neural network layer produces a new correspondence matrix by updating the previous correspondence matrix. This process is applied k times resulting in $\mathbf{S}^{(1)}, \dots, \mathbf{S}^{(K)}$.

For every iteration k , where $k \in \{1, \dots, K\}$, each consensus loop begins by assigning an arbitrary coloring to the nodes of G_s using randomly generated vectors. Let \mathbf{R}_s be a matrix whose i -th row contains the vector of the i -th node of G_s . Then, the model assigns a coloring to the nodes of G_t based on their relationship to the nodes of G_s (as captured by matrix $\mathbf{S}^{(k-1)}$). Thus, matrix \mathbf{R}_t is produced as follows: $\mathbf{R}_t = [\mathbf{S}^{(k-1)}]^\top \mathbf{R}_s$. These colors are distributed across their respective graphs by using synchronous message passing, i.e., by applying $\text{GNN}^{(k)}$ to each graph. That is, the model computes new matrices of features $\mathbf{Z}_s^{(k)} = \text{GNN}^{(k)}(\mathbf{A}_s, \mathbf{R}_s, \mathbf{E}_s)$ and $\mathbf{Z}_t^{(k)} = \text{GNN}^{(k)}(\mathbf{A}_t, \mathbf{R}_t, \mathbf{E}_t)$. By comparing the emerging representations of two nodes, a matrix $\mathbf{D} \in \mathbb{R}^{n_s \times n_t}$ is produced which measures the neighborhood consensus between all pairs of nodes. For example, for the i -th node of G_s and the j -th node of G_t , we have that:

$$\mathbf{D}_{i,j} = [\mathbf{Z}_s^{(k)}]_{i,:} - [\mathbf{Z}_s^{(k)}]_{j,:}$$

where $[\mathbf{Z}]_{i,:}$ denotes the i -th row of matrix \mathbf{Z} . Matrix \mathbf{D} is then transformed using a multi-layer perceptron and the new correspondence matrix is computed as:

$$\mathbf{S}^{(k)} = \text{softmax}(\mathbf{S}^{(k-1)} + \text{MLP}(\mathbf{D}))$$

Instead of considering only two correspondence matrices as in [9], in our setting, the final correspondence matrix \mathbf{S} is computed as the weighted sum of all the soft correspondence matrices $\mathbf{S}^{(0)}, \dots, \mathbf{S}^{(K)}$, where the weights correspond to trainable parameters followed by a final softmax function (applied in a row-wise manner).

$$\mathbf{S} = \text{softmax}\left(\mathbf{w}_0 \mathbf{S}^{(0)} + \mathbf{w}_1 \mathbf{S}^{(1)} + \dots + \mathbf{w}_k \mathbf{S}^{(K)}\right)$$

where $\mathbf{w} \in \mathbb{R}^k$ is a trainable vector. To train the model, we employ the categorical cross-entropy:

$$\mathcal{L} = - \sum_{i=1}^{n_s} \log (\mathbf{S}_{i,\pi(i)})$$

where $\pi(i)$ gives the node of G_t that matches to node i of G_s . Since the loss function is differentiable, we can train the model in an end-to-end fashion using stochastic gradient descent.

4 Experiments

In this section, we report on the experimental evaluation of the proposed method and compare with state-of-the-art methods.

4.1 Datasets

We evaluate the proposed method in the task of keypoint matching in natural images and in the task of geometric feature matching, where only the coordinates of the keypoints are available and not their visual features. For the first task, we experiment with the following two datasets: (i) PASCAL-VOC [8] with Berkeley annotations of keypoints [2]; and (ii) WILLOW-ObjectClass [3]. The PASCAL-VOC dataset is an extension of PASCAL-VOC 2011 and contains annotations of body parts for 20 semantic classes. The Willow ObjectClass dataset consists of 5 categories from Caltech-256 and Pascal VOC 2007. For the second task, we experiment with the PASCAL-PF dataset [14]. The PASCAL-PF dataset consists of annotations for 20 semantic classes. The annotations for each image pair consist of a set of keypoint coordinate pairs where each pair shares the same arbitrary numeric label.

4.2 Baselines

We compare the proposed model against recently proposed neural network models that have achieved stat-of-the-art performance. More specifically, on the PASCAL-VOC and WILLOW ObjectClass datasets, we compare the proposed model against the following three models: (i) GMN [28]; (ii) PCA-GM [26]; and (iii) DGMC [9]. On the PASCAL-PF dataset, we compare our model against two models: (i) GMN [29]; and (ii) DGMC [9]. For the DGMC model, L refers to the number of neighborhood consensus iterations used. For all baselines, we use the results reported in [9].

4.3 Experimental Setup

On all three considered datasets, we follow the experimental setups of previous studies [4, 9, 26, 28], and we use the same training and test splits as in those studies.

For the proposed model, we set the number of epochs to 30, and the batch size to 512. We use the Adam optimizer with a learning rate of 10^{-3} . We set the hidden-dimension size of the graph neural networks to 128. We tune the following two hyperparameters: (i) the number of graph neural networks used in the neighborhood consensus step K ; and (ii) the number of message passing iterations of each graph neural network r . For the proposed model and DGMC, two types of edge features are evaluated: (i) isotropic edge features that use normalized relative distances which result in the neighbors of a node being equally considered during message passing; and (ii) anisotropic edge features that use 2D cartesian coordinates which result in some neighbors having a greater weight during message passing. $\text{GNN}^{(0)}, \dots, \text{GNN}^{(K)}$ all correspond to SplineCNN models [11]. The message passing scheme employed by the SplineCNN model is defined as:

$$\mathbf{h}_i^{(t+1)} = \sigma \left(\mathbf{W}^{(t+1)} \mathbf{h}_i^{(t)} + \sum_{j \in \mathcal{N}(i)} \Phi_{\theta}^{(t+1)}(\mathbf{e}_{i,j}) \mathbf{h}_j^{(t)} \right)$$

Table 1. Hits@1 (%) on the PASCAL-PF and PASCAL-VOC dataset

PASCAL-VOC	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	M-Bike	Person	Plant	Sheep	Sofa	Train	TV	Mean
GMN	31.1	46.2	58.2	45.9	70.6	76.5	61.2	61.7	35.5	53.7	58.9	57.5	56.9	49.3	34.1	77.5	57.1	53.6	83.2	88.6	57.9
PCA-GM	40.9	55.0	65.8	47.9	76.9	77.9	63.5	67.4	33.7	66.5	63.6	61.3	58.9	62.8	44.9	77.5	67.4	57.5	86.7	90.9	63.8
DGMC $L = 20$ Iso	50.1	65.4	55.7	65.3	80.0	83.5	78.3	69.7	34.7	60.7	70.4	59.9	70.0	62.2	56.1	80.2	70.3	88.8	81.1	84.3	68.3
Ours $K = 1$, $r = 4$ Iso	47.8	66.5	57.5	56.9	74.4	85.1	75.7	69.4	33.8	55.9	80.9	63.1	69.5	58.0	49.8	78.6	61.5	90.2	77.6	83.1	66.8
DGMC $L = 20$ Aniso	47.0	65.7	56.8	67.6	86.9	87.7	85.3	72.6	42.9	69.1	84.5	63.8	78.1	55.6	58.4	98.0	68.4	92.2	94.5	85.5	73.0
Ours $K = 1$, $r = 5$ Aniso	52.0	71.5	56.0	72.1	84.6	85.5	83.9	75.7	48.0	70.5	88.1	66.0	85.1	64.5	58.2	96.2	67.5	93.5	93.6	88.7	75.0
PASCAL-PF	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	M-Bike	Person	Plant	Sheep	Sofa	Train	TV	Mean
(Zhang & Lee, 2019)	76.1	89.8	93.4	96.4	96.2	97.1	94.6	82.8	89.3	96.7	89.7	79.5	82.6	83.5	72.8	76.7	77.1	97.3	98.2	99.5	88.5
DGMC $L = 20$	81.1	92.0	94.7	100.0	99.3	99.3	98.9	97.3	99.4	93.4	100.0	99.1	86.3	86.2	87.7	100.0	100.0	100.0	100.0	99.3	95.7
Ours $K = 5$, $r = 4$	82.2	92.2	94.5	99.4	99.3	98.4	98.6	98.2	98.6	94.9	100.0	99.0	85.6	87.2	87.8	99.5	100.0	99.4	99.5	98.9	95.7

Table 2. Average inference and training times on the Pascal-PF and PASCAL-VOC datasets.

PASCAL-VOC	Inference time	Training time (per epoch)
DGMC $L = 20$ Iso	89 ms	29 s
Ours $K = 1$, $r = 4$ Iso	36 ms	17 s
DGMC $L = 20$ Aniso	158 ms	60 s
Ours $K = 1$, $r = 5$ Aniso	94 ms	47 s
PASCAL-PF	Inference time	Training time (per epoch)
DGMC $L = 20$	40 ms	34 s
Ours $K = 5$, $r = 4$	19 ms	36 s

where $\mathcal{N}(i)$ denotes the set of neighbors of the i -th node of the graph, and the trainable B-spline based kernel function $\Phi_\theta(\cdot)$ is conditioned on edge features.

The proposed model is implemented in PyTorch [24] using the PyTorch Geometric library [10] and KeOPs [12]. All experiments were performed on a single machine equipped with NVidia Tesla P100-PCIE GPU.

4.4 Results

Following previous work, we use Hits@1 as our evaluation metric, which measures the proportion of correct matches in the top result. We also measure the running time of the proposed model.

PASCAL-VOC. Table 1 (top) shows the Hits@1 scores achieved by the different methods. We observe that the anisotropic variants of the proposed model and of DGMC outperform all the other models by wide margins. Interestingly, the anisotropic variant of the proposed model is the best performing method since it yields a relative increase of 2% on the mean Hits@1 over DGMC. In terms of running time, as shown in Table 2 (top), the variants of the proposed model are faster than those of DGMC both during training and during inference. This can be of high importance for real-time scenarios where generating predictions in a small amount of time is of a critical nature.

Table 3. Hits@1 (%) on the WILLOW-ObjectClass dataset

Method	Face	Motorbike	Car	Duck	Winebottle
GMM	99.3	71.4	74.3	82.8	76.7
PCA-GM	100.0	76.7	84.0	93.5	96.9
DGMC $L = 20$ Iso	100.00 ± 0.00	92.05 ± 3.24	90.28 ± 4.67	88.97 ± 3.49	97.14 ± 1.83
Ours $r = 3$, $K = 3$ Iso	99.40 ± 0.46	87.78 ± 3.76	83.82 ± 4.09	85.20 ± 2.86	92.53 ± 1.69
DGMC $L = 10$ Aniso	100.00 ± 0.00	98.80 ± 1.58	96.53 ± 1.55	93.22 ± 3.77	99.87 ± 0.31
Ours $r = 4$, $K = 3$ Aniso	99.85 ± 0.24	99.35 ± 0.71	97.88 ± 1.09	93.17 ± 2.67	99.15 ± 0.73

Table 4. Average inference and training times on the WILLOW-ObjectClass dataset.

Method	Inference time	Training time (per Epoch)
DGMC L = 20 Iso	57 ms	13 s
Ours $r = 3$, $K = 3$ Iso	37 ms	8 s
DGMC L = 10 Aniso	64 ms	23 s
Ours $r = 4$, $K = 3$ Aniso	51 ms	16 s

WILLOW-ObjectClass. Tables 3 and 4 illustrate the performance (Hits@1 score) and running time of the different models, respectively. The anisotropic variant of DGMC outperforms all the other methods, while the anisotropic variant of the proposed model is the second best method and is slightly outperformed by DGMC. Surprisingly, the isotropic instance of our model achieves much smaller Hits@1 score than that of DGMC, while it is even outperformed by PCA-GM. In both cases, the proposed model is trained in a smaller amount of time than DGMC, while it is also more efficient at inference time. In Fig. 1, we also provide some examples where the proposed model has successfully find correspondences between keypoints extracted from pairs of images.

PASCAL-PF. Table 1 (bottom) reports the Hits@1 score achieved by the proposed model and the baselines. We observe that the proposed model matches the performance of DGMC, while both models outperform the model proposed in [29]. This indicates that our model is very effective even if visual features are not available. Table 2 (bottom) illustrates the running time of the proposed model and that of DGMC. The proposed model’s average inference time per image is much smaller than that of DGMC, while training times of the two models are similar to each other.

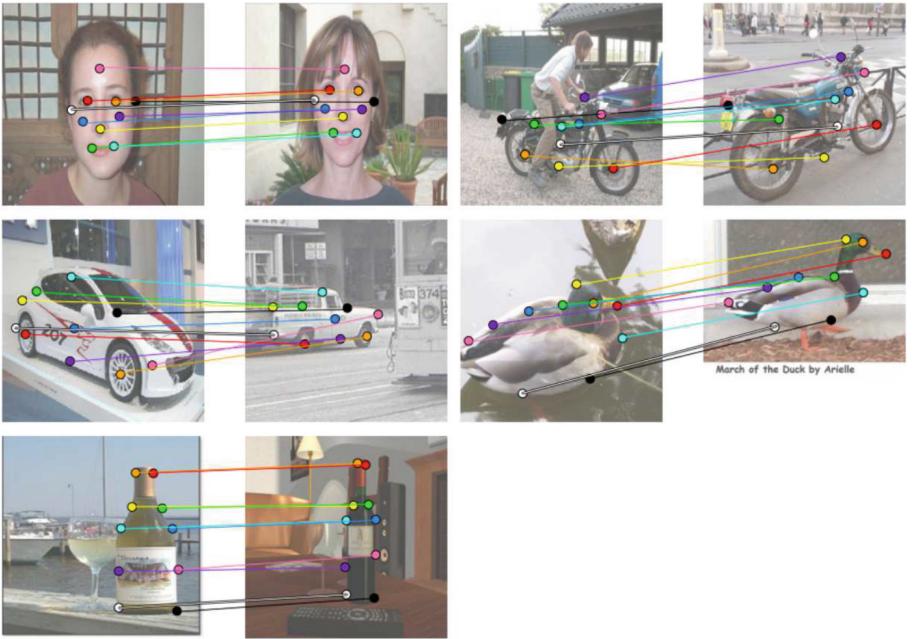


Fig. 1. Examples of matching keypoints from the WILLOW-ObjectClass dataset.

5 Conclusion

In this paper, we presented a graph neural network for the problem of image matching. Our starting point is a previous work on graph neural networks for graph matching [9], which we modify to use not only two correspondence matrices, but all the matrices that are produced during the consensus stage. Furthermore, we also replace the iteration in the neighborhood consensus step with a series of graph neural networks. Experiments on standard image matching datasets showed that the proposed model is more efficient than the baseline model while it maintains and in some cases improves prediction performance.

Acknowledgements. This research is co-financed by Greece and the European Union (European Social Fund- ESF) through the Operational Programme «Human Resources Development, Education and Lifelong Learning» in the context of the project “Reinforcement of Postdoctoral Researchers - 2nd Cycle” (MIS-5033021), implemented by the State Scholarships Foundation (IKY). This work was also supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP).

References

1. Bai, Y., Ding, H., Bian, S., Chen, T., Sun, Y., Wang, W.: Simgnn: a neural network approach to fast graph similarity computation. In: Proceedings of the 12th ACM International Conference on Web Search and Data Mining, pp. 384–392 (2019)

2. Bourdev, L., Malik, J.: Poselets: body part detectors trained using 3d human pose annotations. In: International Conference on Computer Vision (ICCV) (2009). <http://www.eecs.berkeley.edu/~lbourdev/poselets>
3. Cho, M., Alahari, K., Ponce, J.: Learning graphs to match. In: 2013 IEEE International Conference on Computer Vision, pp. 25–32 (2013)
4. Choy, C.B., Gwak, J.Y., Savarese, S., Chandraker, M.: Universal correspondence network. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, pp. 2414–2422 (2016)
5. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. *Int. J. Pattern Recognit. Artif. Intell.* **18**(03), 265–298 (2004)
6. Cuturi, M.: Sinkhorn distances: lightspeed computation of optimal transport. *Adv. Neural Inf. Process. Syst.* **26**, 2292–2300 (2013)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 886–893 (2005)
8. Everingham, M., Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision* **88**(2), 303–338 (2010). <https://doi.org/10.1007/s11263-009-0275-4>
9. Fey, M., Lenssen, J.E., Morris, C., Masci, J., Kriege, N.M.: Deep graph matching consensus. In: International Conference on Learning Representations (ICLR) (2020)
10. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch Geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds (2019)
11. Fey, M., Lenssen, J.E., Weichert, F., Müller, H.: SplineCNN: fast geometric deep learning with continuous B-spline kernels. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
12. Feydy, J., Glauñès, J., Charlier, B., Bronstein, M.: Fast geometric learning with symbolic matrices. In: Advances in Neural Information Processing Systems, vol. 33 (2020)
13. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 1263–1272. PMLR (2017)
14. Ham, B., Cho, M., Schmid, C., Ponce, J.: Proposal flow. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
15. Heimann, M., Shen, H., Safavi, T., Koutra, D.: Regal: representation learning-based graph alignment. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 117–126 (2018)
16. Kim, S., Min, D., Ham, B., Jeon, S., Lin, S., Sohn, K.: FCSS: fully convolutional self-similarity for dense semantic correspondence. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2017)
17. Knopp, P., Sinkhorn, R.: Concerning nonnegative matrices and doubly stochastic matrices. *Pac. J. Math.* **21**(2), 343–348 (1967)
18. Lee, J., Kim, D., Ponce, J., Ham, B.: Sfnet: learning object-aware semantic correspondence. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2273–2282 (2019)
19. Liu, C., Yuen, J., Torralba, A.: Sift flow: dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(5), 978–994 (2011)
20. Loiola, E.M., de Abreu, N.M.M., Boaventura-Netto, P.O., Hahn, P., Querido, T.: A survey for the quadratic assignment problem. *Eur. J. Oper. Res.* **176**(2), 657–690 (2007). <https://www.sciencedirect.com/science/article/pii/S0377221705008337>

21. Long, J., Zhang, N., Darrell, T.: Do convnets learn correspondence? In: Proceedings of the 27th International Conference on Neural Information Processing Systems, NIPS'14, vol. 1, p. 1601–1609. MIT Press, Cambridge (2014)
22. Lowe, D.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157 (1999)
23. Ma, J., Jiang, X., Fan, A., Jiang, J., Yan, J.: Image matching from handcrafted to deep features: a survey. *Int. J. Comput. Vision* **129**(1), 23–79 (2021)
24. Paszke, A., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d’ Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32, pp. 8024–8035. Curran Associates, Inc. (2019). <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
25. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperGlue: learning feature matching with graph neural networks. In: CVPR (2020). <https://arxiv.org/abs/1911.11763>
26. Wang, R., Yan, J., Yang, X.: Learning combinatorial embedding networks for deep graph matching. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3056–3065 (2019)
27. Xu, H., Luo, D., Zha, H., Duke, L.C.: Gromov-Wasserstein learning for graph matching and node embedding. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 6932–6941. PMLR (2019)
28. Zanfir, A., Sminchisescu, C.: Deep learning of graph matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2684–2693 (2018)
29. Zhang, Z., Lee, W.S.: Deep graphical feature learning for the feature matching problem. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5087–5096 (2019)



Relations Between Entropy and Accuracy Trends in Complex Artificial Neural Networks

Lucia Cavallaro^{1(✉)}, Marco Grassia², Giacomo Fiumara³, Giuseppe Mangioni², Pasquale De Meo⁴, Vincenza Carchiolo⁵, Ovidiu Bagdasar¹, and Antonio Liotta⁶

¹ University of Derby, Kedleston Road, Derby DE22 1GB, UK
{l.cavallaro,o.bagdasar}@derby.ac.uk

² Dipartimento di Ingegneria Elettrica, Elettronica e Informatica,
Università degli Studi di Catania, Catania, Italy
{marco.grassia,giuseppe.mangioni}@unict.it

³ MIFT Department, Università degli Studi di Messina, Messina 98166, Italy
g.fiumara@unime.it

⁴ Università degli Studi di Messina, Polo Universitario Annunziata,
Messina 98122, Italy
p.demeo@unime.it

⁵ Dipartimento di Matematica e Informatica, Università degli Studi di Catania,
Catania, Italy
vincenza.carchiolo@unict.it

⁶ Free University of Bozen-Bolzano, Faculty of Computer Science, Bolzano, Italy
antonio.liotta@unibz.it

Abstract. Training Artificial Neural Networks (ANNs) is a non-trivial task. In the last years, there has been a growing interest in the academic community in understanding how those structures work and what strategies can be adopted to improve the efficiency of the trained models. Thus, the novel approach proposed in this paper is the inclusion of the entropy metric to analyse the training process. Herein, indeed, an investigation on the accuracy computation process in relation to the entropy of the intra-layers' weights of multilayer perceptron (MLP) networks is proposed. From the analysis conducted on two well-known datasets with several configurations of the ANNs, we discovered that there is a connection between those two metrics (*i.e.*, accuracy and entropy). These promising results can be helpful in defining, in the future, new criteria to evaluate the training process goodness in real-time by optimising it and allow faster detection of its trend.

Keywords: Complex artificial neural networks · Network science · Graph theory · Entropy · Accuracy

1 Introduction

Deep Learning [1], the sub-branch of Machine Learning that uses *deep* models, is now pervasive in many fields and has been successfully employed to approach many complex tasks [2–4]. The motivations behind its popularity are well known: the large availability of data and computational power, and the capability of the trained models to generalise to unseen data, often with super-human performances. However, the training of Deep Learning models is far from trivial. In fact, various attempts are often required to find a combination of the hyperparameters (*e.g.*, the number of model layers, the number of neurons, the number of training epochs, etc.) such that the model is able to learn from the data and yet does not *overfit* (*i.e.*, the model performs a good generalisation to new data). Some works in the literature address this problem and aim to simplify the training phase, for instance by using Genetic Algorithms to select the model topology [5–7]. On the other hand, other works aim at accelerating the training by using sparse Artificial Neural Network models: the first work in this direction is by *Mocanu et al.* [8], who proposes an evolutionary algorithm that takes inspiration from biological neural networks, but others followed [9, 10].

In this preliminary work we aim at analysing the weights of fully connected Artificial Neural Network models to better understand their training using Network Science tools. In fact, whereas the application of Deep Learning algorithms to Complex Networks has quite a long history (from the seminal works of *Scarselli et al.* [11] to the rise of Geometric Deep Learning [12]), the vice-versa, (*i.e.*, the application of Network Science techniques to get insights from Deep Learning models) is still largely unexplored.

Herein, in particular, we borrow from Network Science the entropy metric, nowadays widely used to measure the complexity of complex networks such as in the works of *Gómez-Gardeñes et al.* [13] and *Bianconi et al.* [14, 15], even though its main applications has been in information theory field [16, 17].

Our intuition has been to combine the analysis of the accuracy on the validation set during the training process with the intra-layer entropy weights to detect whether there is a relation between those two metrics. In order to do so, we tested two well-known datasets (*i.e.*, Lung cancer and COIL20) and we set up multilayer perceptron networks (MLP) in supervised training fashion in which we varied the number of hidden layers and neurons per each hidden layers from which we, then, discarded the underfitting and overfitting configurations. Herein, for the sake of brevity, only the slow and fast learning have been reported.

The results obtained confirmed our intuition. Indeed, what we discovered is that the entropy is growing during the transitory phase of the accuracy and starts to drop sharply once the accuracy became stable. Thus, there is a connection between those two metrics. This outcome paves the way in devising a new metrics that, combined with the classical Machine Learning techniques, may be proposed as a powerful tool to speed up training process of ANNs.

The paper is organised as follows. Section 2 describes all the tools required to understand the research herein presented. In particular, it is structured in theoretical background (Sect. 2.1), description of the datasets used (Sect. 2.2),

and methodology followed (Sect. 2.3). Next, in Sect. 3, the results obtained are shown and commented jointly with the conclusions.

2 Materials and Methods

In this section all the basic notions and definitions used in this work are showed jointly with a concise description of the datasets used in our experiments along with the methodology followed to pursue the experiments.

2.1 Background

In this paper, we deal with *Artificial Neural Networks* (ANNs) in their classical configuration, which means that they are fully connected networks.

An ANN is a computational learning system that uses the structure and the functions of a network with the goal of understanding and converting a certain data input of one form into a desired output, usually in another form. This concept has been originated from several studies on human biology and the way neurons of the human brain work together. In particular, we focus our attention on a *multilayer perceptron* (MLP) supervised model.

MLP is a *feed-forward* ANN composed by several hidden layers, forming a Deep Network. The name of feed-forward derives from the structure of those kind of models because information flows through the function being evaluated from x , through the intermediate computations used to define a function f , and finally to the output y . In addition, there are no feedback connections in which outputs of the model are fed back into themselves. It is, however, possible to have feedback connections in a feed-forward neural network; in such a case, those networks called recurrent neural networks [18]. Feed-forward neural network are used for classification and regression, as well as for pattern encoding [19]. For the sake of brevity, herein, we consider only classification problems.

Supervised learning involves observing several samples of a given dataset, which will be divided into *training* and *test* samples. While the former is used to train the neural network, the latter works as a litmus test, as it is compared with the ANN predictions [1, 9, 18]. Indeed, thanks to the training set, the supervised learning models are able to teach models to obtain the desired output to the ANN. Thus, the training dataset is composed by inputs and correct outputs that allow the model to learn over time (*i.e.*, epochs). During the training process, hence, the algorithm can validate its accuracy through the loss function, adjusting until the error has been sufficiently minimised. Other types of learning strategies are unsupervised learning and semi-supervised learning.

Generally speaking, one of the most popular evaluation metrics to verify the goodness of the training process is performed by computing the accuracy, which measure how the predicted values are close to the expected value. It is computed as follows:

$$\text{Accuracy} = \frac{\text{pred}_{\text{correct}}}{\text{pred}_{\text{tot}}} \quad (1)$$

where $\text{pred}_{\text{correct}}$ is the number of correct predictions and pred_{tot} is the total number of predictions.

Due to the strict connection between the notion of ANN and networks, those systems can be seen as weighted graphs. A *weighted graph*, denoted as G , is a triplet $G = \langle V, E, W \rangle$ in which V is the set of *vertices* (or nodes), $E = \{\langle i, j \rangle : i \in V \wedge j \in V\}$ is the set of *edges* (or links) and $W : E \rightarrow \mathbb{R}^+$ is a function that maps an edge $\langle i, j \rangle$ onto a non-negative real number w_{ij} . In addition, a *signed graph* is a weighted graph where if $w_{ij} \in \mathbb{R}$ (*i.e.*, weights can also be negative). Lastly, if a graph's weights are equal only to zero or one; *i.e.*, $w_{ij} = 1$, $w_{ij} = 0$, then it is called *unweighted graph*.

With those notions in mind, we employ the Shannon's entropy S to get an insight about the distributions. It is defined as follows:

$$S = - \sum_{i \in W} p_i \log p_i \quad (2)$$

where

$$p_i = \frac{w_i}{\sum_{j \in W} w_j}$$

and i represents the index of the i -th weight, whereas j is the index of all the elements in the set. Note that $i, j \in E_l$, then both are in the edges set of the specific layer l .

This formulation of *entropy* is a bit counter-intuitive: one would expect that the higher the entropy, the more uniform the distribution. The reason is that if weights are evenly distributed the p_i will all have different values, while if all weights are equal all the p_i are also equal. For instance, if there are only two weights $w_1 = 1$ and $w_2 = 2$, then $p_1 = 1/3$, $p_2 = 2/3$. On the other hand, if $w_1 = w_2 = 1$, then $p_1 = p_2 = 1/2$. After translating the weights into the p_i s, this formulation should get more intuitive.

In addition, the maximum value (the upper bound) of entropy is equal to $\log N$, where N represents the number of elements in the set. In fact:

$$p_i = \frac{1}{N} \quad \forall i$$

and, thus,

$$S = - \sum p_i \log p_i = -N \cdot \left(\frac{1}{N}\right) \cdot \log \left(\frac{1}{N}\right) = \log N$$

An introductory discussion on the concept of information entropy is contained in [20], where entropy (and mutual information) are used to detect the key nodes of a complex network. More details on the use of entropy in complex networks can be found on the work of Alves *et al.* [21], in which the authors analysed the evolution over time of this metric in the context of the global value chain of the worldwide production networks. For a thorough description of the various definitions of the entropy metrics in complex network we refer the interested reader to [22].

2.2 Dataset

To conduct the experiments, two publicly available online¹ well-known datasets have been used.

The first one is the LUNG CANCER² [23] that is a biological dataset composed by features on lung cancer to train the ANN to detect this specific kind of cancer, which contains 3312 gene data obtained from 17 people with normal lungs and 186 lung cancer patients that is classified into 5 classes: (i) Adenocarcinomas (139 patients), (ii) Squamous Cell Lung Carcinomas (21 patients), (iii) Pulmonary Carcinoids (20 patients), (iv) Small Cell Lung Carcinomas (6 patients), and (v) Normal Lung (17 people).

The second one is COIL20³ [24, 25] that is an image dataset used to train ANNs to detect 20 different objects. The images of each object were taken five degrees apart as the object is rotated on a turntable and each object has 72 images. The size of each image is 32×32 pixels, with 256 grey levels per pixel. Thus, each one is represented by a 1024-dimensional vector [9].

The number of instances, input features and output classes of both datasets are summarised in Table 1.

Table 1. Dataset structures description. From left: dataset name; number of instances, of input features and of output classes.

Name	Instances (#)	Input features (#)	Output classes (#)	Source
Lung cancer	203	3,312	5	[23]
COIL20	1440	1024	20	[24, 25]

2.3 Methodology

In this paper we want to address to what extent there are similarities between the behaviours of accuracy (on the validation set) and entropy in the artificial networks under scrutiny. In this respect, we tested different models with a variable number of hidden layers (from 2 to 3) and neurons for each layer (*i.e.*, 2, 5, 10, 50, 100, 250, 500) from which we detected the best models for the datasets under scrutiny and discarded the underfit and overfit scenarios. For the sake of brevity, the four most significant configurations obtained, which are summarised in Table 2, are herein reported. Lastly, 100 epochs for the training have been considered.

During the training, we have, then, computed the entropy of weights of the inter-layer links of the nodes (*i.e.*, neurons) of each layer. Note that the entropy has been computed for the absolute values of the weights, since in an Artificial

¹ <http://featureselection.asu.edu/>.

² <https://sites.google.com/site/feipingnie/file/https://jundongl.github.io/scikit-feature/datasets.html>.

³ <http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html>.

Table 2. Configurations of hidden layer and neurons per each one. From left: dataset name, number of hidden layers, number of neurons per hidden layer and learning speed.

Dataset	Hidden layers (#)	Neurons (#)	Learning speed
Lung Cancer	L2	N10	Slow
Lung Cancer	L2	N50	Fast
Lung Cancer	L3	N10	Slow
Lung Cancer	L3	N50	Fast
COIL20	L2	N10	Slow
COIL20	L2	N50	Fast
COIL20	L3	N10	Slow
COIL20	L3	N50	Fast

Neural Network weights can also be negative and entropy is defined only for non-negative values. Finally, the analysis for each layer to layers pair is reported aggregated (*i.e.*, the analysis of the whole model).

3 Discussion and Conclusions

In this section the results obtained from our analysis are commented. In Figs. 1 and 2 are shown the comparative trend of accuracy and entropy for Lung and COIL20 datasets, respectively.

Note that to normalise the entropy outcomes, its values have been reported as the ratio between the current entropy value and its maximum theoretical value (*i.e.*, $\log N$, that is the upper bound).

First of all, as expected, in the slow learning scenarios (Figs. 1a, 1c, 2a, and 2c) the highest accuracy attained does not reach a plateau greater than 90%. It is fascinating to notice that a higher number of hidden layers not necessarily leads to a more precise training. In Fig. 2d, whereas the configuration selected has three hidden layers and 50 neurons per layer, for instance, the accuracy reached at the last epoch (and all the previous values before that one as well) is lower than the one reported in Fig. 2b in which there is the same number of neurons, but only two hidden layers.

What emerges from our analysis is that the more stable the accuracy, the higher the entropy decrease. This aspect becomes clearer when Fig. 2a and Fig. 2c are compared with the other plots. Indeed, in those two figures, which represent specifically the slow learning trend in COIL20 dataset, the accuracy is still increasing without reaching its plateau; thus, the entropy is increasing as well. On the other hand, once the accuracy is stable (*i.e.*, no more improvements in terms of performances are expected at this step of the training process) the entropy starts to decrease. For instance, in Fig. 1d after the 15th epoch, the trend of the accuracy tends to stabilise and, at the same time, the entropy starts to sharply drop.

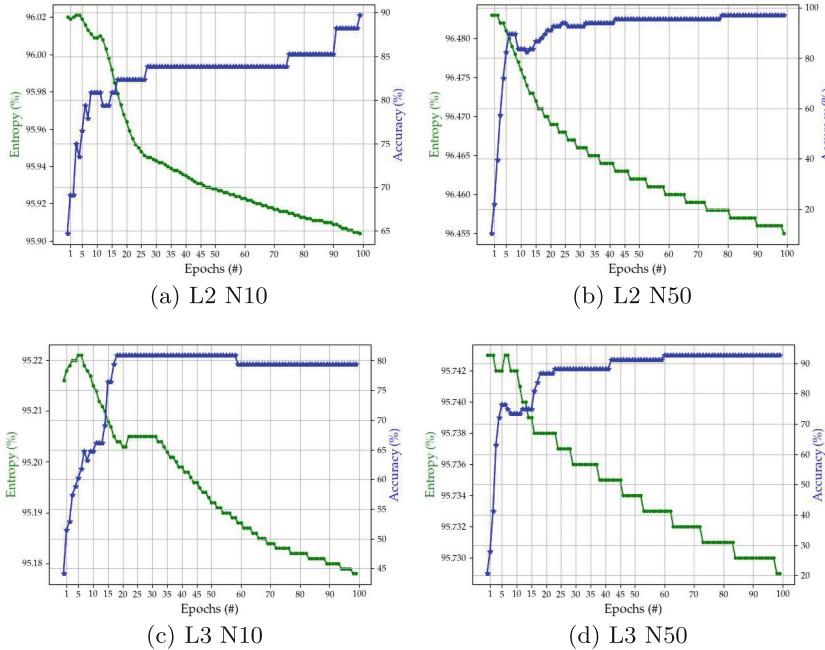


Fig. 1. Comparative analysis of Entropy (dotted markers in green) and Accuracy (star markers in blue) on Lung Dataset

Another interesting aspect to pinpoint is that the fluctuations experienced by the accuracy are somehow followed by the entropy. Of course, all the variations in the entropy are significantly less pronounced respect to the accuracy trend; that is, the entropy ratio varies between 93% and 97% whereas the accuracy starts from 0% up to 95%. Nonetheless, the trend is still remarkable.

What we infer from those analyses is that the entropy variations are somehow related with the accuracy. This is an aspect that can pave the way to a branch of studies on this direction.

One of them can be to extend this work considering a wider range of datasets and configurations to detect whether there are specific datatets on which this approach is preferred. Thus, it would be also matter of study varying not only the number of hidden layers, but also the number of neurons per hidden layer that can be also unbalanced (instead of keeping it constant for all the hidden layers under scrutiny).

Other possibilities can be the use of different learning strategies (*i.e.*, unsupervised, semi-supervised).

Another application domain to involve the use of entropy for a finer tuning of the training process can be in Sparse ANNs. Indeed, we are also interested in understanding how does entropy behave as the network becomes sparser. Indeed, the advantage of combining the entropy with the accuracy could lead to

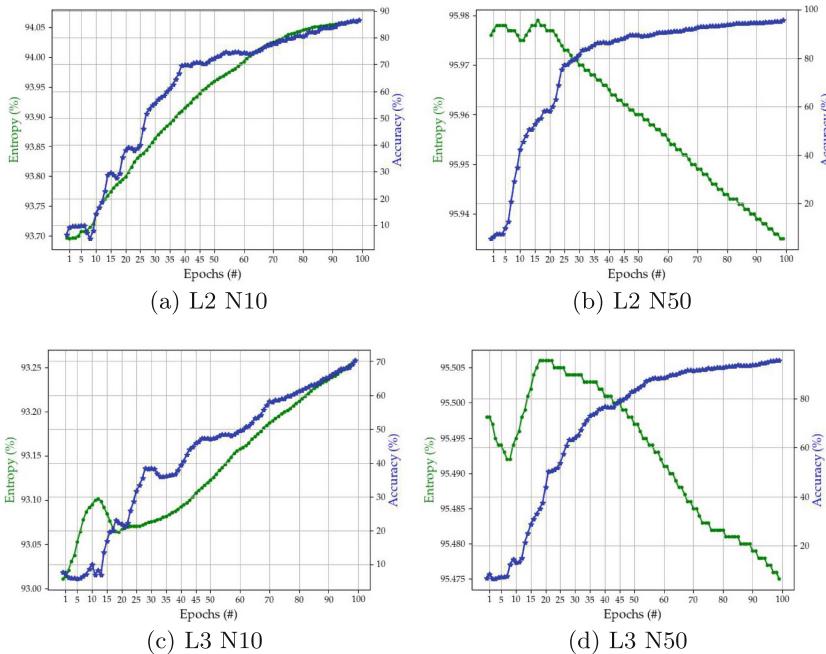


Fig. 2. Comparative analysis of Entropy (dotted markers in green) and Accuracy (star markers in blue) on COIL20 Dataset

more precise metric, which could be used, in turn, to develop a better and more powerful tool able to define a suitable stop condition for ANNs training. This could lead to a reduction in terms of the overall number of epochs needed to train the network with a satisfying accuracy and, thus, have a lower computational cost.

References

1. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
2. Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F.E.: A survey of deep neural network architectures and their applications. *Neurocomputing* **234**, 11–26 (2017)
3. Jumper, J., et al.: Highly accurate protein structure prediction with AlphaFold. *Nature* **96**(7873), 583–589 (2021)
4. Grassia, M., De Domenico, M., Mangioni, G.: Machine learning dismantling and early-warning signals of disintegration in complex systems. *Nature Commun.* **12**, 5190 (2021)
5. Young, S.R., Rose, D.C., Karnowski, T.P., Lim, S.-H., Patton, R.M.: Optimizing deep learning hyper-parameters through an evolutionary algorithm. In: Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments, pp. 1–5 (2015)

6. Assunção, F., Lourenço, N., Machado, P., Ribeiro, B.: Evolving the topology of large scale deep neural networks. In: Castelli, M., Sekanina, L., Zhang, M., Cagnoni, S., García-Sánchez, P. (eds.) EuroGP 2018. LNCS, vol. 10781, pp. 19–34. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-77553-1_2
7. Mattioli, F., Caetano, D., Cardoso, A., Naves, E., Lamounier, E.: An experiment on the use of genetic algorithms for topology selection in deep learning. *J. Electr. Comput. Eng.* **2019**, 1–12 (2019)
8. Mocanu, D.C., Mocanu, E., Stone, P., Nguyen, P.H., Gibescu, M., Liotta, A.: Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Commun.* **9**(1), 1–12 (2018)
9. Cavallaro, L., Bagdasar, O., De Meo, P., Fiumara, G., Liotta, A.: Artificial neural networks training acceleration through network science strategies. *Soft Comput.* **24**(23), 17787–17795 (2020)
10. Liu, S., Ni'mah, I., Menkovski, V., Mocanu, D.C., Pechenizkiy, M.: Efficient and effective training of sparse recurrent neural networks. *Neural Comput. Appl.* **33**, 1–12 (2021)
11. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. *IEEE Trans. Neural Netw.* **20**(1), 61–80 (2008)
12. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Process. Mag.* **34**, 18–42 (2017)
13. Gómez-Gardeñes, J., Latora, V.: Entropy rate of diffusion processes on complex networks. *Phys. Rev. E* **78**, 065102 (2008)
14. Bianconi, G., Coolen, A.C.C., Perez Vicente, C.J.: Entropies of complex networks with hierarchically constrained topologies. *Phys. Rev. E* **78**, 016114 (2008)
15. Bianconi, G.: Entropy of network ensembles. *Phys. Rev. E* **79**, 036114 (2009)
16. Shannon, C.E.: A mathematical theory of communication. *Bell Syst. Tech. J.* **27**(3), 379–423 (1948)
17. Cover, T., Thomas, J.: Elements of Information Theory, pp 33–36. Wiley, New York (1991)
18. Goodfellow, I.J., Bengio, Y., Courville, A., Deep Learning. MIT Press, Cambridge (2016). <http://www.deeplearningbook.org>
19. Gori, M.: Chapter 5 - deep architectures. In: Gori, M., (ed.) Machine Learning, pp. 236–338. Morgan Kaufmann (2018)
20. Li, Y., Cai, W., Li, Y., Du, X.: Key node ranking in complex networks: a novel entropy and mutual information-based approach. *Entropy* **22**(1), 52 (2020)
21. Alves, L.G.A., Mangioni, G., Rodrigues, F.A., Panzarasa, P., Moreno, Y.: Unfolding the complexity of the global value chain: strength and entropy in the single-layer, multiplex, and multi-layer international trade networks. *Entropy* **20**(12), 909 (2018)
22. Omar, Y.M., Plapper, P.: A survey of information entropy metrics for complex networks. *Entropy* **22**(12), 1417 (2020)
23. Bhattacharjee, A., et al.: Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proc. Natl. Acad. Sci.* **98**(24), 13790–13795 (2001)
24. Cai, D., He, X., Han, J., Huang, T.S.: Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(8), 1548–1560 (2010)
25. Cai, D., He, X., Han, J.: Speed up kernel discriminant analysis. *VLDB J.* **20**(1), 21–33 (2011)



Curved Markov Chain Monte Carlo for Network Learning

John Sigbeku¹, Emil Saucan², and Anthea Monod^{1(✉)}

¹ Imperial College London, London SW7 2AZ, UK

a.monod@imperial.ac.uk

² ORT Braude College of Engineering, Karmiel 2161002, Israel

Abstract. We present a geometrically enhanced Markov chain Monte Carlo sampler for networks based on a discrete curvature measure defined on graphs. Specifically, we incorporate the concept of graph Forman curvature into sampling procedures on both the nodes and edges of a network explicitly, via the transition probability of the Markov chain, as well as implicitly, via the target stationary distribution, which gives a novel, curved Markov chain Monte Carlo approach to learning networks. We show that integrating curvature into the sampler results in faster convergence to a wide range of network statistics demonstrated on deterministic networks drawn from real-world data.

Keywords: Discrete curvature · Graph Forman curvature · Markov chain Monte Carlo · Network learning · Network sampling

1 Introduction

The availability of abundant datasets present many new opportunities for insight into many fields of application, such as biology, social and telecommunications, and sensor analysis. However, the computational burden of fully characterizing a network grows significantly with the size of the dataset and stands to hinder these potential insights. A natural approach is to sample the larger networks to obtain a smaller representation for analysis, which drives the importance of obtaining an accurate representation by selecting the most pertinent characteristics of the network. In this way, network sampling is an important means for learning networks.

Recent work incorporates the inherent geometry of the network in the sampling process, where the discrete curvature of the network has been shown to improve sampling efficiency and accuracy in a deterministic setting [2]. The idea is based on the observation that discrete curvature captures a notion of information flow on the network and thus is a natural concept to exploit when sampling networks with the aim to learn them. Here, we adapt this approach to the random setting, which then makes statistical techniques of random sampling applicable to the problem of network learning. Specifically, we present a curvature-based sampling approach based on the Markov chain Monte Carlo

(MCMC) technique. Our approach incorporates the *graph Forman curvature* of a network into the transition probabilities of a Markov chain defined on the nodes of a network as well as into the target stationary distribution. In this way, our proposed methodology is both explicit and implicit. We demonstrate our approach on deterministic networks arising from real-world data. Results are bench-marked against the uniform Markov sampler to identify the cases where curvature-based sampling outperforms the base case. We show that our geometrically enhanced MCMC sampler provides a quicker convergence to a wide set of network statistics.

The remainder of this paper is organized as follows. We close this section with a brief overview of related work. In Sect. 2, we present the notion of graph Forman curvature; we also briefly overview the concept of MCMC sampling. In Sect. 3, we present our graph Forman-curved MCMC sampler for both edges and nodes. We also give the network statistics that will serve as network characteristics to retain in the sampled subnetwork. Sect. 4 presents details on the implementation and evaluation of our method; we demonstrate our approach on real-world deterministic networks. We conclude with a discussion and opportunities for future research in Sect. 5.

Related Work. Relevant to the ideas presented in this paper, edge- and node-based sampling methods, such as random edge and random node algorithms, draw from the statistical aspect of random sampling to sample representative networks [6]. Similarly, Markov chains on the nodes of the network with a specific stationary distribution have also been proposed, such as the Metropolis–Hastings random walk with uniform stationary distribution [7].

Existing sampling algorithms commonly assume full knowledge of the entire network, requiring its storage on local computer memory which can be computationally costly. To bypass this expense, network crawling algorithms working on the basis of restricted access, where only the current node and its neighbors are required, have also been proposed, such as the forest fire sampling algorithm which visits a neighbor node with pre-specified probability [6].

2 Overview: Discrete Curvature and Random Sampling

We now give a concise overview of the two main concepts which underlie the methodology we propose in this paper, namely, graph Forman curvature and Markov chain Monte Carlo.

2.1 Discrete Curvature on Networks: Graph Forman Curvature

Methods to handle data driven by geometric notions have recently inspired and established a variety of research domains such as topological data analysis, which aims to study the shape of data, and geometric deep learning, which incorporates geometric invariance into learning algorithms. In computational settings, discretization is a fundamental consideration, giving rise to discrete geometry

and, in particular, discrete curvature. In this work, we study *graph Forman curvature* [10, 12], which is the adaptation of the Forman–Ricci curvature [4] to the edges of a network.

For an edge $\langle i, j \rangle \in E$, graph Forman curvature takes into account its own weight, w_{ij} ; the weights of the nodes it connects, $w(i)$ and $w(j)$; and the weights of incident edges excluding the edge itself $\langle i, j \rangle$, $e(i) \sim \langle i, j \rangle$ and $e(j) \sim \langle i, j \rangle$.

Definition 1. Given a network $G = \{V, E\}$, the graph Forman curvature $F(\cdot)$ for an edge $\langle i, j \rangle \in E$, is given by

$$F(\langle i, j \rangle) = w_{ij} \cdot \left(\frac{w(i)}{w_{ij}} + \frac{w(j)}{w_{ij}} - \sum_{e(i) \sim \langle i, j \rangle} \frac{w(i)}{\sqrt{w_{ij} w_{e(i)}}} - \sum_{e(j) \sim \langle i, j \rangle} \frac{w(j)}{\sqrt{w_{ij} w_{e(j)}}} \right). \quad (1)$$

Here, the edges in highly concentrated areas connecting nodes which are also connected to many other nearby nodes are given the highest absolute Forman curvature.

In the combinatorial case, where each edge and node weight is set to one, (1) reduces to

$$F(\langle i, j \rangle) = 4 - d(i) - d(j), \quad (2)$$

where $d(i)$ and $d(j)$ are the degrees of the nodes connected by $\langle i, j \rangle$. For further details, see [9, 10].

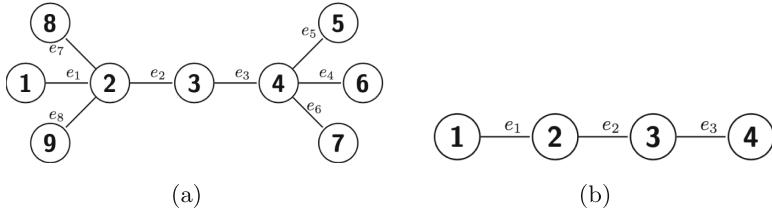


Fig. 1. Example networks to illustrate graph Forman curvature. (a) A combinatorial network; (b) A zero Forman curvature network.

Example 1. Figure 1(a) illustrates how the graph Forman curvature captures the concentrated areas of a network. From (2), we have $F(e_2) = F(e_3) = -2$, since both edges connect nodes with degrees two and four. On the other hand, $F(e_1) = F(e_4) = F(e_5) = F(e_6) = F(e_7) = F(e_8) = -1$ which all connect nodes with degrees one and four. Edges e_2 and e_3 connect nodes with greater degrees than any of the remaining edges, and so they represent more concentrated areas of the network.

Example 2. Figure 1(b) illustrates zero graph Forman curvature, assigned to any edge in a line-shaped combinatorial network where each connected node has degree of two, given by edge e_2 here. This corresponds to the intuitive notion of curvature, where zero curvature corresponds to flatness.

Moreover, graph Forman curvature can be extended onto the nodes of network by summing the individual curvatures of the edges incident to a node. For a node i , this can be written as

$$F(i) = \sum_{e(i)} F(\langle i, k \rangle), \quad (3)$$

where $e(i)$ are the edges incident to node i . In our work, we apply both node- and edge-based Forman curvature to our sampling methodology.

2.2 Random Sampling: Markov Chain Monte Carlo

Probability-based inference entails estimating the expected value of a statistic or density from a probabilistic model, which can often be intractable in complex and high-dimensional settings. Approximation procedures have been developed to overcome these difficulties; one such procedure is Markov chain Monte Carlo (MCMC).

In the network setting where there is inherent dependence, the statistical sampling procedure applied must necessarily take the dependence structure into account. MCMC is a well-established technique comprising a class of algorithms to draw samples from high-dimensional and complex distributions that allows for a dependence structure in sampling, where the next sample drawn takes into account the existing sample; see e.g., [8].

Briefly, Monte Carlo allows for random sampling from a probability distribution to approximate a quantity or statistic of interest. These methods typically sample independently and assume efficiency of sampling from the desired *target* or *stationary distribution*. A Markov chain is a sequence of random variables X_1, X_2, \dots where each random variable in the sequence depends only on the variable drawn prior. Given this dependence, there is then a specified *transition probability* between each sequential pair of random variables,

$$\mathbb{P}(X_{n+1} = x \mid X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) = \mathbb{P}(X_{n+1} = x \mid X_n = x_n).$$

Merging the two concepts gives rise to the technique of MCMC, which is used to perform inference for probability distributions where independent samples cannot be drawn.

In this setting, Monte Carlo cannot be directly applied, so samples are drawn from the probability distribution by constructing a Markov chain. With enough iterations, the chain will eventually settle, or find equilibrium, on our quantity of interest (target distribution). Since the purpose of the sample is still to approximate a quantity of interest, it technically is a Monte Carlo sample. Thus, MCMC sampling generates a Monte Carlo sample that is drawn from Markov chain, which incorporates probabilistic dependence.

3 Forman-Curved Markov Chain Monte Carlo

For a given network $G = \{V, E\}$, we are interested in obtaining a sample sub-network $G_n = \{V', E'\}$ with $V' \subseteq V$ and $E' \subseteq E$. We would like for this

subnetwork to capture pertinent characteristics of the larger network in order to serve as a faithful representation. In the context of MCMC, we are interested in estimating these characteristics, which take the form of *network statistics*, by sampling. The intuition behind incorporating the geometry of the network is that the additional information will improve the sampling method, either by obtaining more accurate estimates or quicker convergence to the target value of interest, i.e., the network statistics. In particular, we incorporate graph Forman curvature into the sampling procedure, so that the sequence of random variables in the Markov chain are driven by curvature of the network. This idea can be considered intuitively as regions of high curvature driving the direction of Markov chain.

In this section, we propose two methods of incorporating graph Forman curvature into network sampling: an explicit *edge-based* approach, and an implicit *node-based* approach. We also give the network statistics that our proposed sampling method aims to approximate.

3.1 the Edge-Based Approach

The edge-based Forman-curved MCMC sampler that we now propose explicitly integrates curvature into the transition probability of a Markov chain.

Using MCMC, G_n may be obtained from the Markov chain on the set of nodes V that is the sequence of random variables X_1, X_2, \dots, X_n , where the actualizations are nodes, and the transition probability between nodes is given by

$$\begin{aligned} p_{ij} &= \mathbb{P}(X_{k+1} = j \mid X_1 = i_1, X_2 = i_2, \dots, X_k = i) = \mathbb{P}(X_{k+1} = j \mid X_k = i) \\ &\propto \frac{|F(\langle i, j \rangle)|}{d(j)} \end{aligned} \quad (4)$$

for all $i, j \in V$, and where $F(\langle i, j \rangle)$ is given by (1).

Notice that this Markov property holds for all k , making this Markov chain time-homogeneous. Also, two graph samples G_n and G_{n+1} are identical if the node X_{n+1} has already been sampled, even though their Markov chains differ in length. The chain can be initialized at different starting nodes X_0 to obtain different chains and graph samples.

3.2 The Node-Based Approach

We now shift the focus from integrating curvature to alter transition probabilities of the Markov chain which generates node samples, to generating samples from a target probability distribution defined on the nodes of the network. This approach implicitly incorporates curvature by defining a target distribution based on the graph Forman curvature around the nodes of the network. In particular, a Forman-curved distribution where nodes are sampled with probability

$$\mathbb{P}(X_k = i) \propto \frac{|F(i)|}{d(i)} \quad (5)$$

for large enough k , where $F(i)$ is given by (3). This target distribution can be thought of as our desired stable state, where in the long run as we continue to sample, we tend towards drawing node i $100 \cdot \mathbb{P}(X_k = i)$ times per hundred draws.

The Metropolis–Hastings (MH) algorithm is an MCMC algorithm often used in Bayesian statistics to generate samples from a specified target distribution up to a constant of proportionality; it is useful when the normalizing factor for a distribution is unknown. It is implemented by sampling a known proposal probability distribution q and applying a corrective procedure. In our setting, we have the additional benefit of employing an MH algorithm to generate samples from the graph Forman-curved target distribution by crawling the network.

Algorithm 1. Metropolis–Hastings for Networks

```

1: input Unnormalized density  $g$ ; proposal distribution  $q$ ; starting node  $X_0$ 
2: for  $k = 1, 2, \dots$  do
3:   Set  $Y_k \sim q(X_{k-1})$ 
4:   Set  $U \sim \text{Unif}(0, 1)$ 
5:   if  $U \leq \min\left(\frac{g(Y_k)q(Y_k, X_{k-1})}{g(X_{k-1})q(X_{k-1}, Y_k)}, 1\right)$  then
6:      $X_k = Y_k$ 
7:   else
8:      $X_k = X_{k-1}$ 
9:   end if
```

In our implementation of Algorithm 1, g is the target distribution with $g(i) = |F(i)|/d(i)$ and the proposal distribution is given by $q(X_{k-1}, Y_k) = 1/d(i)$ when $X_{k-1} = i$ so that each neighbor node is proposed with equal probability.

In this node-based approach, the graph Forman curvature can be thought to implicitly influence the transition probability of the Markov chain, since the algorithm implies

$$p_{ij} = \begin{cases} q_{ij} \cdot \frac{g(j)q_{ji}}{g(i)q_{ij}} & \text{if } \frac{g(j)q_{ji}}{g(i)q_{ij}} \leq 1; \\ q_{ij} & \text{otherwise.} \end{cases}$$

In contrast to the edge-based approach, this approach requires full access of the network to first calculate the target distribution. This is because the full set of network weights are required to calculate Forman curvature around all nodes in the network (i.e., $\{F(1), F(2), \dots, F(|V|)\}$). If the target distribution is not calculated beforehand, full access will be required once the Markov chain reaches a node that is connected to all other remaining nodes in the network, since evaluating the transition probabilities will require the set $\{F(1), F(2), \dots, F(|V|)\}$.

In both approaches, the degree is used as the normalization factor for the transition probabilities in order to bias sampling towards nodes with larger Forman curvature on average, in the combinatorial sense.

3.3 Network Statistics

Network statistics capture characteristics of the network in order to assess the faithfulness in representation of the sampled network in reference to the original larger network. In this work, we focus on *centrality* network statistics, which describe different notions of node importance. We apply our proposed Forman-curved MCMC samplers given above in Sects. 3.1 and 3.2 to estimate the following network statistics. We specify a wide variety of network statistics in order to study how well our sampling procedure is able to capture different characteristics of the larger network, e.g., [1].

Definition 2. For a given network, let σ_{ij} be the number of shortest paths $P(\cdot, \cdot)$ from node i to j and $\sigma_{ij}(i')$ be the number of those shortest paths passing through node i' . The betweenness centrality $BC(\cdot)$ of a node i' is given by

$$BC(i') = \sum_{i \neq i' \neq j} \frac{\sigma_{ij}(i')}{\sigma_{ij}}.$$

The closeness centrality $CC(\cdot)$ of a node i is given by $CC(i) = \frac{1}{\sum_{j \neq i} P(i, j)}$.

Intuitively, betweenness centrality captures how much a node acts as a bridge between groups of nodes in a network; it captures how important a node is in connecting separate groups of nodes together. Closeness centrality measures how close one node is to any other node in the network.

Definition 3. Let $W = (W_{ij}) \in \mathbb{R}^{|V| \times |V|}$ be the weighted adjacency matrix of a network. The strength $s(\cdot)$ of a node i is given by the sum of the rows or columns of the matrix W ,

$$s(i) = \sum_{j=1}^{|V|} W_{ij} = \sum_{j=1}^{|V|} W_{ji}.$$

Notice that the strength of a node is a weighted version of the degree of a node and therefore can be seen as a weighted, localized measure of node importance.

Definition 4. Let $A = (A_{ij}) \in \{0, 1\}^{|V| \times |V|}$ be the adjacency matrix of a network and W its weighted version as above. The weighted clustering coefficient (e.g., [3]) is given by

$$c^w(i) = \frac{\sum_{j=1}^{|V|} \sum_{h=1}^{|V|} \{W_{ij} + W_{ih}\} A_{ij} A_{ih} A_{jh}}{2s(i)(d(i) - 1)}.$$

While the standard clustering coefficient measures the proportion of possible connections between the neighbors of a given node, the weighted version takes into account the relative importance of the clustering structure around a given node by incorporating the edge weights.

4 Implementation and Results

We implemented our proposed Forman-curved MCMC samplers given above in Sects. 3.1 and 3.2 to approximate the network statistics given in Sect. 3.3.

Convergence performance for the curved samplers was compared to the base case of uniform sampling, where the transition to each neighbor node is with equal probability $p_{ij} = 1/d(i)$ for the edge-based approach and the uniform target distribution $\mathbb{P}(X_k = i) = 1/|V|$ for the node-based approach. Uniform sampling corresponds to standard MCMC approaches where curvature is not integrated into the method.

4.1 Performance Evaluation

To evaluate the performance of our proposed method, we compared the convergence of the network sample $G_n = \{V', E'\}$ to the full network G by examining convergence to global network statistics across multiple Markov chains initialized at different starting points.

More specifically, for a given network statistic, let Z denote its random variable for the full network. Then for all network samples $n = 1, 2, \dots, N$ and all chains $c = 1, 2, \dots, n_c$, we examined the convergence of the estimator for the mean of each network statistic distribution

$$\bar{Z}_{nc} = \frac{1}{|V'|} \sum_{j=1}^{|V'|} Z_{nc}(j)$$

by calculating the mean squared error $MSE_n = \frac{1}{n_c} \sum_{c=1}^{n_c} \{\bar{Z}_{nc} - \mathbb{E}[Z]\}^2$.

4.2 Application: Deterministic Real-World Networks

We applied our curved edge-based and node-based MCMC samplers to deterministic networks arising from real-world data. Specifically, we studied the character interaction network from the novel “Les Misérables” by Victor Hugo, with 77 nodes and maximum degree of 36 obtained from The KONECT Project [5] and the neural network of the *Caenorhabditis elegans* (*C. elegans*) worm with 306 nodes and maximum degree of 134 [11].

We implemented both the edge- and node-based curved Forman MCMC samplers proposed in Sects. 3.1 and 3.2 to estimate all four network statistics listed in Sect. 3.3 on the Les Misérables network; the results are displayed in Figs. 3 and 4, respectively. Examples of sampled subnetworks are displayed in Fig. 2. We implemented the node-based curved Forman MCMC sampler proposed in Sect. 3.2 to estimate all four network statistics listed in Sect. 3.3 on the *C. elegans* network; the results are displayed in Fig. 5.

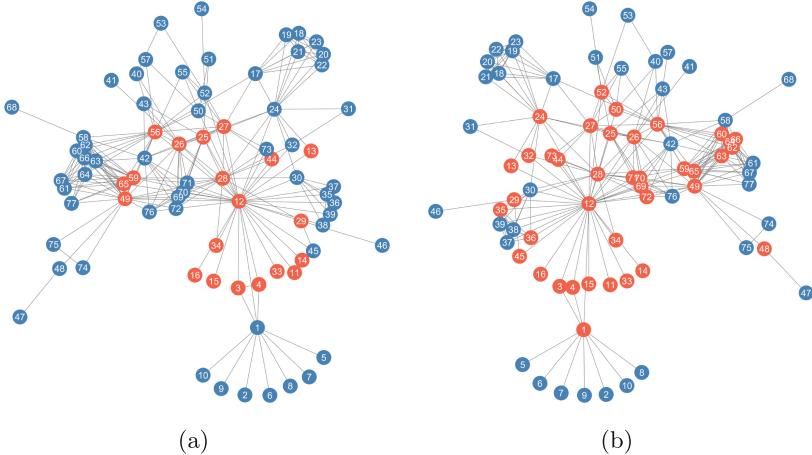


Fig. 2. Les Misérables network: The blue nodes represent all nodes in the full network, while the orange nodes represent subsampled nodes by our proposed method. (a) Top 25% sampled nodes; (b) Top 50% sampled nodes.

Across all sampling methods and networks, there is a visible advantage in integrating curvature into the MCMC sampling scheme. In particular, the Forman-curved sampler outperforms the uniform sampler when estimating mean strength and clustering coefficient. These results imply that the local behavior of graph Forman curvature is compatible with that of these network statistics.

For the two Forman-curved approaches applied to the Les Misérables network, we see similar convergence behavior to the uniform case for mean betweenness centrality. For the edge-based curved sampler, we see an initial advantage in incorporating curvature which is then overtaken by the uniform sampler. Overall, edge-based curved sampling converges faster than node-based curved sampling as shown in Fig. 3 compared to Fig. 4. This is likely due to the uncorrelated node samples for the edge-based curved method and the fact that the methods view uniformity from two different levels. Furthermore, it is in concordance with the fact that for networks, Ricci curvature is an edge-based measure.

There appears to be some evidence that mean betweenness centrality is better estimated by the curved Forman sampler for larger networks and for small sample sizes, which would benefit small sample reconstruction. Further experiments on larger networks will need to be tested in order to corroborate this observation.

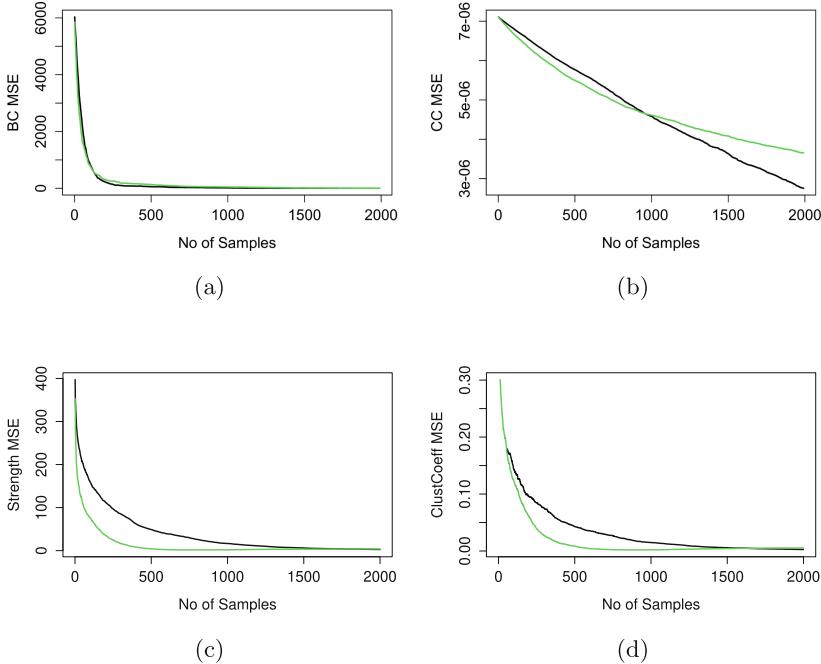


Fig. 3. Les Misérables network: Edge-based Forman-curved MCMC. Black curve: Uniform sampling; Green curve: Sampling with transition probability (4). (a) Mean Betweenness Centrality; (b) Mean Closeness Centrality; (c) Mean Strength; (d) Mean Clustering Coefficient.

5 Discussion

In this work, we studied the problem of network learning by sampling. The impetus is that a larger network may be studied by an appropriate representation obtained by sampling; the retained network characteristics in the sample are determined by a wide variety of network statistics. We estimated these statistics from the probabilistic viewpoint of Markov chain Monte Carlo and proposed a novel sampler that incorporates the discrete curvature of the network both explicitly and implicitly via both edge- and node-based approaches. Specifically, curvature was integrated into the transition probability of the Markov chain as well as the target stationary distribution. We found that our curved samplers noticeably improves the convergence to network statistics comprising local behavior.

A particular advantage of our method is that graph Forman curvature is a local measure and thus requires only local information on the current node and its immediate neighbors. In this sense, the technique can be viewed as semi-crawl-like and therefore is still applicable even without access to the full network.

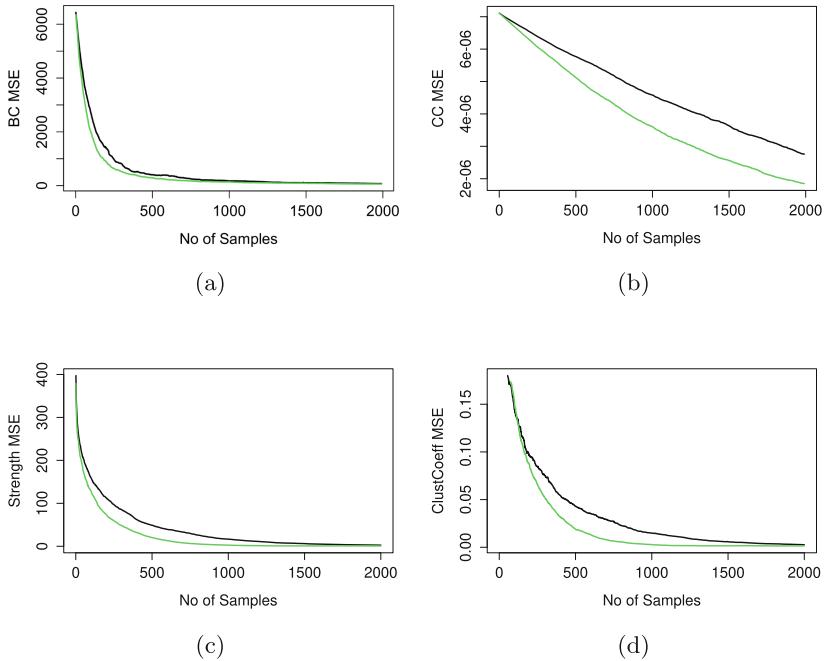


Fig. 4. Les Misérables network: Node-based Forman-curved MCMC. Black curve: Uniform sampling; Green curve: Sampling node i with probability (5). (a) Mean Betweenness Centrality; (b) Mean Closeness Centrality; (c) Mean Strength; (d) Mean Clustering Coefficient.

This opens up possibilities for future research to develop semi-supervised network learning methodology by incorporating discrete curvature notions into reversible jump MCMC.

Other natural directions of research include studying the accuracy of the estimated network statistics. Other modifications to our samplers could also be studied; for instance, normalization by strength, which would result in bias towards nodes with largest weighted average, is a natural extension. Additionally, it would be interesting to compare graph Forman–Ricci curvature with other types of Ricci curvature for networks as in [2] and other geometrically motivated network measures. Furthermore, additional experimentation on large-scale, real-world networks would help achieve a better understanding of the performance and applicability of our method on these scales, entailing further refinements of algorithms to increase their efficiency.

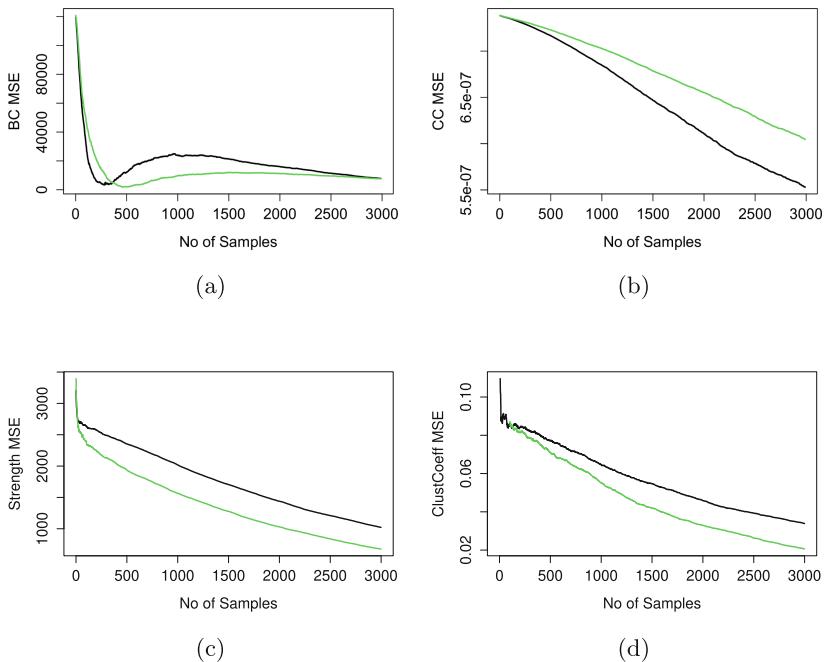


Fig. 5. *C. elegans* network: Node-based Forman-curved MCMC. Black curve: Uniform sampling; Green curve: Sampling node i with probability (5). (a) Mean Betweenness Centrality; (b) Mean Closeness Centrality; (c) Mean Strength; (d) Mean Clustering Coefficient.

References

1. Airololi, E.M., Bai, X., Carley, K.M.: Network sampling and classification: An investigation of network model representations. *Decision Support Systems* **51**(3), 506–518 (2011)
2. Barkanass, V., Jost, J., Saucan, E.: Geometric sampling of networks. arXiv preprint [arXiv:2010.15221](https://arxiv.org/abs/2010.15221) (2020)
3. Barrat, A., Barthélémy, M., Pastor-Satorras, R., Vespignani, A.: The architecture of complex weighted networks. *Proc. Natl. Acad. Sci.* **101**(11), 3747–3752 (2004)
4. Forman, R.: Bochner's method for cell complexes and combinatorial Ricci curvature. *Discrete Comput. Geom.* **29**(3), 323–374 (2003). <https://doi.org/10.1007/s00454-002-0743-x>
5. J. Kunegis. KONECT – the Koblenz network collection. In: Proceedings of the International Conference on World Wide Web on World Wide Web Companion, pp. 1343–1350 (2013)
6. Leskovec, J., Faloutsos, C.: Sampling from large graphs. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 631–636 (2006)
7. Li, R.-H., Yu, J.X., Qin, L., Mao, R., Jin, T.: On random walk based graph sampling. In: 2015 IEEE 31st International Conference on Data Engineering, pp. 927–938. IEEE (2015)

8. Murphy, K.P.: Machine Learning: A Probabilistic Perspective. MIT Press, Cambridge (2012)
9. Saucan, E., Samal, A., Weber, M., Jost, J.: Discrete curvatures and network analysis. MATCH Commun. Math. Comput. Chem. **20**(1), 605–622 (2018)
10. Sreejith, R.P., Mohanraj, K., Jost, J., Saucan, E., Samal, A.: Forman curvature for complex networks. J. Stat. Mech. Theor. Exp. 063206 (2016). <https://iopscience.iop.org/article/10.1088/1742-5468/2016/06/063206>
11. Watts, D.J., Strogatz, S.H.: Collective dynamics of ‘small-world’ networks. Nature **393**(6684), 440–442 (1998)
12. Weber, M., Jost, J., Saucan, E.: Forman-Ricci flow for change detection in large dynamics data sets. Axioms **5**(4), 26 (2016)



Extracting Semantic Information from Dynamic Graphs of Geometric Data

Devavrat Vivek Dabke^(✉) and Bernard Chazelle

Princeton University, Princeton, NJ 08544, USA
ddabke@princeton.edu

Abstract. In this paper, we demonstrate the utility of dynamic network sequences to provide insight into geometric data; moreover, we construct a natural syntactic and semantic understanding of these network sequences for useful downstream applications. As a proof-of-concept, we study the trajectory data of basketball players and construct “interaction networks” to express an essential game mechanic: the ability for the offensive team to pass the ball to each other. These networks give rise to a library of player configurations that can in turn be modeled by a jump Markov model. This model provides a highly compressed representation of a game, while capturing important latent structures. By leveraging this structure, we use a Transformer to predict trajectories with increased accuracy.

Keywords: Geometric data · Networks · Dynamic networks · Machine learning

1 Introduction

Multi-agent systems are fascinating both for their geometric properties and for their complex interactions. In a variety of contexts, we would like to understand their underlying dynamics, moving beyond the construction of black-box models that simply replicate their behavior. Thus we strive to produce a model that leverages dynamic networks (i.e. networks with a topology that changes over time) to encode a geometric system. We can then extract “semantic” information about these network sequences to crystallize our understanding of the underlying dynamics. One paragon example of a multi-agent geometric system includes fast-paced “invasion” sports like basketball, soccer, and hockey [8].

We study the trajectories of players as they move across the court. To develop a rich understanding of the dynamics of basketball players, we develop a model with:

1. Formation discovery: a semantic understanding of the functional roles of players;
2. High compression: an efficient representation of a game, as player trajectory data is large and difficult to interpret;
3. Predictive power: a mechanism for generating synthetic basketball data and predicting trajectories of players.

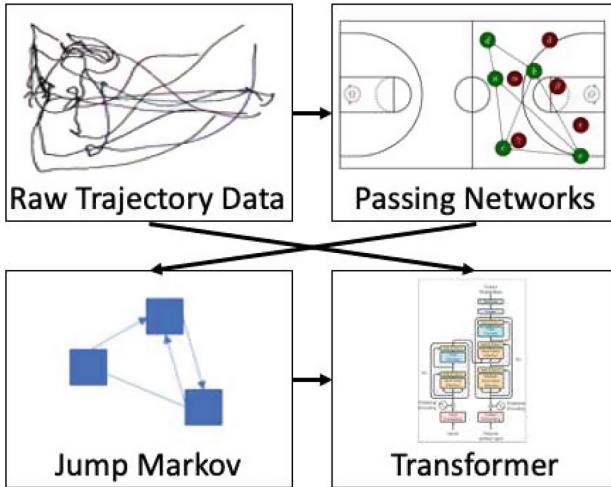


Fig. 1. Overview of the data analysis pipeline. First, the raw trajectory data is converted into interaction networks. Second, by comparing graphs up to isomorphism, we can construct a “library” of possible configurations. We can then construct a jump Markov model by taking the empirical maximum likelihood estimator with graphs as the state space. Finally, we can feed in the raw trajectory data and the graph data from the jump Markov model into a Transformer model for prediction. This article opens the door to future work on inferring game semantics and strategies from actual games.

In summary, the synthesis of semantic and geometric data is the main contribution of this paper, which is a theoretical innovation, as well as a practical one: on an important downstream task (trajectory prediction), we greatly improve on past results. We propose a novel pipeline to convert geometric data from a complex, interactive multi-agent system into semantic sequences. This new perspective provides better insight into the underlying dynamics, as well as stronger results in important applications like trajectory prediction.

2 Related Work

The analysis of basketball player trajectory began shortly after cutting-edge technology was developed that allowed for comprehensive player tracking [11]. Early models did reasonably well in role discovery and compression. Over the past decade, however, predictive power has greatly increased, especially for the most common prediction task, which is trajectory prediction.

2.1 Trajectory Prediction

Trajectory prediction focuses on forecasting the movement of players given their history. Generically, trajectory prediction is a much broader, and relatively old, discipline, but, for concreteness, we restrict ourselves to those that involve sports [13, 15].

Generally, state-of-the-art techniques involve black-box machine learning models that predict K frames of data from L initial frames [7, 10, 18]. Some models explicitly construct roles for players, while others avoid this complication. Whether or not explicit formation or role discovery is important to trajectory prediction is not well-established, however.

2.2 Role Discovery

Discovering a player’s function on the court is an interesting challenge. The naive approach of tracking players by their personal identity across plays, games, seasons, and teams may yield a more confused analysis. Therefore, one common approach to analyzing sports data, especially with trajectories, is to develop some kind of “role” categorization, e.g. a point guard.

Work first done in [11], extended in [19], patented in [3], and updated in [9] proposes an extensive set of efficient methods for classifying roles, especially within field hockey. Role discovery is an important strand of research, as it emphasizes the semantics of the game. While trajectory prediction is a compelling problem in its own right, role discovery highlights the underlying structure of a particular game. Through role discovery, we can provide interpretable labels or classifications to particular player formations and movements.

Role discovery has also garnered popular attention [1, 2, 4, 12, 20], with a variety of approaches in constructing and classifying roles.

2.3 Network Analysis

Network analysis for sports data is comparatively old, with some early efforts in soccer beginning in 1979 [6]. The most relevant type of network analysis, however, has been on *passing networks* and investigates the frequency with which players pass the ball [5, 14]. This research direction emphasizes the study of aggregate network properties, e.g. the *centrality* of a player on a particular team [8]. Network analysis thus far has considered the network of passing frequency over an entire game rather than the specific dynamics during the game itself.

3 Semantic Geometric Pipeline

3.1 Geometric Data

Our principal dataset contains the position of the offense, the defense, and the ball—expressed as (x, y) -coordinates—across an entire game capture at 25 frames per second (i.e. 40ms between frames). Notably, basketball is divided into *possessions*, where the teams alternate between defensive and offensive roles. Consider five indexed points as offensive players $O = \{o_1, o_2, o_3, o_4, o_5\} \subset \mathbb{R}^2 : |O| = 5$ and five indexed points as defensive players $D = \{d_1, d_2, d_3, d_4, d_5\} \subset \mathbb{R}^2 : |D| = 5$. One frame F of data is the ordered pair (O, D) . Each possession P is a sequence of frames of data and a game is a sequence of possessions.

3.2 Dynamic Passing Networks

From this dataset, we construct *dynamic passing networks*. These networks are defined over the offense (i.e. for basketball, there are five nodes); two players on the same team have an edge joining them if there is no defender in-between. Figure 2 provides some intuition for this construction.

More precisely, given two sets of five offensive players $\mathcal{O} \subset \mathbb{R}^2$ and five defensive players $\mathcal{D} \subset \mathbb{R}^2$, we can define a graph G over vertices \mathcal{O} . For $o_i, o_j \in \mathcal{O}$, edge (o_i, o_j) is in the graph if there is no $d \in \mathcal{D}$ such that the *line of sight* $l(o_i, o_j)$ intersects with *occlusion field* $F_r(d)$. The line of sight between two points x, y is the line segment joining them. An occlusion field of radius r at point p is the corresponding l_2 -ball centered at p (n.b. $r = 3$, to represent the average 3-foot radius of basketball players).

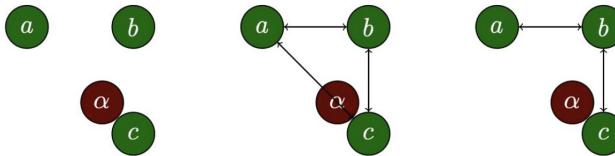


Fig. 2. A deeper look at a snapshot of a basketball game. Green nodes with Roman characters are offensive players. Red nodes with Greek letters are defensive players. On the left, we see player positions with occlusion fields. In the center, we see all offensive lines of sight. On the right, we see the occlusion network with only the offensive edges.

For each frame, we construct these networks by starting with a complete graph with the offensive players as the nodes; we then remove edges from this graph if a defensive player occludes the straight line of sight between a pair of offensive ones. Figure 3 depicts a passing network for one frame.

We convert each frame of the basketball game from two sets of (x, y) -coordinates to a graph. We further compress this representation by only considering graphs up to isomorphism, which allows us to store a label to a representative graph per frame. This procedure thus converts a sequence of frames of position data into a sequence of labels.

Notably, this representation sheds the direct geometry of the basketball game. This sequence of labels provides a purely semantic and highly compressible representation of the game and is justified by three physical assumptions, which are validated by our results:

1. From one frame to the next, there can be at most one edge that changes and this edge change can only occur as a result of well-behaved player trajectories;
2. It is possible for two different geometries to produce the same graph, a sequence of graphs must come from a real play, and thus provide enough information on the possession;
3. Basketball is “fast-paced” enough that it can be assumed to follow a Markovian property. Mainly, players do not have time to consider the history of the game to factor into a future strategy, and instead either follow a set strategy or respond nearly instantaneously to their current environment.

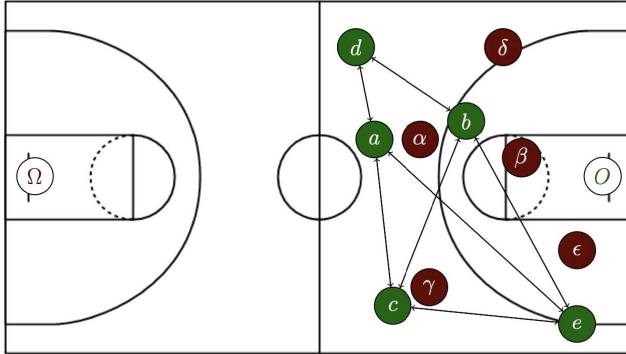


Fig. 3. A frame of the basketball game with the constructed passing network. Green circles with latin letters are offensive players, while red circles with greek letters are defensive players.

3.3 Jump Markov Model

We construct the maximum likelihood estimator of a *jump Markov model* that characterizes this sequence of labels. A Markov model assumes that a sequence of labels exhibits the Markovian property, namely that the label of one entry in the sequence is only influenced by the previous entry and no others. Such a sequence can be characterized by “transition” probabilities: for every possible pair of labels (x, y) , the frequency with which y appears right after x in our sequences; we call this a (time-homogeneous, discrete, finite) Markov chain, which is a stochastic process E_k .

A twist on such a model is a “jump” Markov model that additionally assigns a *hold time* to each state: namely, since a basketball game is a continuous-time process, we can also capture the average length of time that our sequence of frames does not change. More precisely, consider a Poisson process, which is a continuous-time stochastic process with rate λ and associated counting process $\{N(t)\}$. Overlapping the discrete-time Markov chain on the continuous-time Poisson process (where the “events” that occur are transitions of the Markov chain) yields a jump Markov model, defined as the continuous-time stochastic process $X(t)$ such that

$$X(t) = E_{N(t)}$$

which is our stochastic process of interest [16].

3.4 A Transformer

Finally, to validate the empirically constructed jump Markov model, we use a Transformer, which is a state-of-the-art deep neural network that excels at two tasks: sequence completion and sequence translation. Transformer is generally a staple of natural language processing, but can work in a variety of different sequence-related tasks [17]. For our experiments, we retained all of the standard architectural elements of the out-of-the-box Transformer, only making a mild alterations as necessary (described in the following sections).

For our experiments, we divided the frames into short sequences of length 50; with a standard 80-10-10 allocation, these sequences were then split into a training, validation, and test set respectively.

Baseline: Sequence Completion. To establish a baseline, we first used Transformer Decoder for sequence completion: given 40 frames of positional data, we used Transformer to predict 10 frames of positional data. For this task, we removed the standard lookup embedding layer for transformer, and instead directly concatenated all player positions to construct a vector in $\mathbb{R}^{10 \times 2} = \mathbb{R}^{20}$ (10 players, each with an x and y coordinate).

Comparison: Sequence Translation. For our second task, we leveraged Transformer to “translate” between the sequences produced in the state space of the jump Markov model to positional data. In particular, we first converted positional data into passing networks. Then, we compared all of these networks up to isomorphism to create a “library” of possible networks; each possible network was assigned a unique token. We could thus convert sequences of passing networks into sequences of tokens. Finally, feeding this sequence of tokens into Transformer yielded predicted positional data, which we compared against the original raw trajectories. This setup corresponds directly to Fig. 1.

4 Results

4.1 Markovian Property

First, we consider the sequence of graphs that we construct from the geometric data. Understanding the dynamics of these graphs provides insight into the overall dynamics of the game. It is key to find a suitable model that captures this behavior. Fortunately, our data seems to express a Markovian property and we can therefore use a Markov model to capture the essential elements of our graph sequences. In this section, we provide empirical validation for this claim.

To begin with, we recall our assumption that dynamic passing networks can only change by one edge at a time. From a theoretical perspective, we could imagine that, with a high enough sampling frequency, a tie would be unlikely; and, we could also enact some tie-breaking scheme. Empirically, 99.3% of changes in number of edges are within one edge, which we could bring to 100% if we could sample more frequently. In fact, given that 99.3% is quite close to saturation, we can conclude that our sampling frequency is nearly correct: neither too often nor too sparse.

This assumption, that only one edge changes at one time, is convenient, as it allows us to study the change in number of edges of the graphs, which is much simpler to analyze. In other words, we can convert a sequence of graphs into a sequence of number of edges and learn much about our system without having to rely on a full classification of the graphs. We state a fundamental assumption: if the number of edges does not

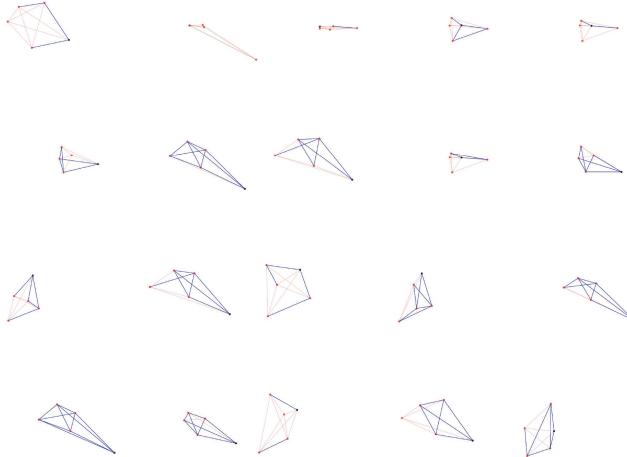


Fig. 4. Sample library of graphs for a possession, descending in order of frequency from left to right and top to bottom.

change in a graph sequence, then the graph (modulo isomorphism) does not change either; this fact follows directly from our assumption that only edge can change at a time. Figure 5 indicates the number of edges in the passing network sequence over time (i.e. per frame).

If we model the number of edges as a Markov chain, we can verify the Markovian property of our edge count data by checking how a change in edges predicts the next change. The edge change probabilities over an entire game are given in Table 1. This data for the game accurately reflects the distribution for each possession, as well, which obviates some concerns about variability within possessions.

Table 1. Probability table for changes in number of edges. The first row indicates that if the previous change was a decrease in number of edges, then there was a 7%, 87%, and 6% chance respectively that the next change in number of edges was a decrease, no change, or increase. The second row follows the same pattern given the previous change was no change in the number of edges. The third row is the same for an increase in number of edges.

	-	0	+
-	0.07	0.87	0.06
0	0.05	0.89	0.05
+	0.07	0.85	0.08

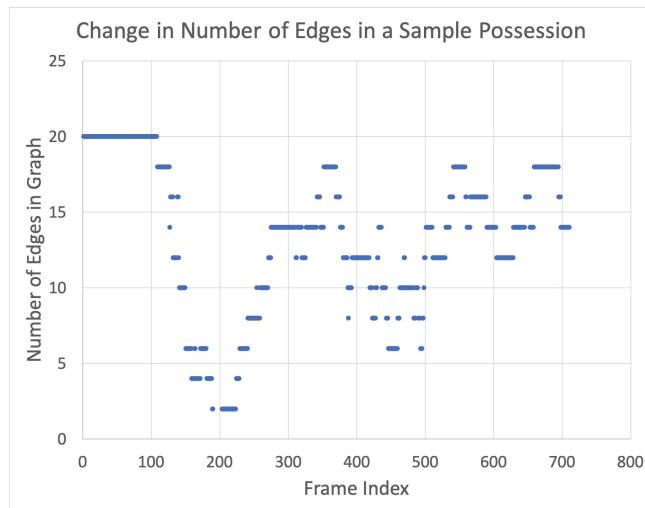


Fig. 5. The number of edges for a particular possession in the game. Mathematically, the number of edges are a random walk over the frames of data within a possession.

The data in Table 1 shows us that the previous change in edges does not influence a future change in edges, in that edge change increments are in fact independent. This data verifies that the number of edges over the course of a possession is in fact Markovian, which is a useful yet surprising property.

Moreover, this suggests that the graph sequence itself is Markovian because of our underlying assumption that one edge change at a time can occur. Therefore, we can use a Markov model to approximate the sequence of graphs over the course of a possession. Finding the maximum likelihood estimator of a Markov chain is also quite simple, as it is given by the empirical transition probabilities.

4.2 Jump Markov Model

While interpreting the sequences of graphs as a Markov chain is a big step towards understanding the underlying dynamics, minor adjustments help us refine the model. In particular, we observe from Table 1 that the overwhelming behavior is for the number of edges in the graph sequences, and hence the sequences themselves, not to change.

Labeling the graph sequences up to isomorphism and constructing the transition probability matrix reveals that the majority of the density is placed on the diagonal, which indicates a self-transition. This behavior warrants some additional investigation. To wit, a basketball game is obviously continuous, but our Markov chain with a transition matrix is discrete. For this reason, we lift the Markov model into a jump Markov model, which allows for different “hold times” per state, i.e. the distribution of time spent in a particular state before a transition occurs. In the naive Markov chain, these hold times are modeled as self-transitions, whereas in a jump Markov model, these are modeled as occurring along some exponential distribution.

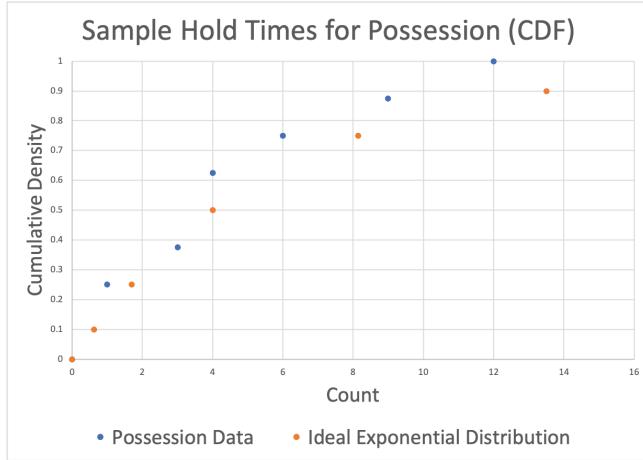


Fig. 6. Sample hold time distribution as a cumulative density function of a representative graph in a representative possession. The blue dots are the true data, which have an average of 5, giving an exponential distribution with parameter $\frac{1}{5}$. The orange dots represent an ideal exponential distribution with the same parameter.

Figure 6 provides a sample hold time distribution for a representative graph in a typical possession. In this case, we see the various hold times for the particular labeled graph, and for reference, an ideal exponential distribution. It appears that a jump Markov model with explicit hold times per graph provides a more robust perspective on state changes.

A jump Markov model displays the stable configurations of players throughout the game. The probability transition matrix for the model suggests the “semantics” of the dynamics, insofar as we can track likely changes in configurations for the players. The hold times demonstrate the overall “stability” of a configuration, i.e. how much time is spent in a particular conformation.

We also can generate a “library” of graphs that provide insight into which configurations appear most often and what typical sequences of graphs look like. Figure 4 is such a representative library that is presented in descending order by frequency of occurrence in a possession.

4.3 Translating with Transformer

Our final set of experiments comes from using a Transformer model. We attempt a standard task in invasion sports, namely predicting the trajectory of player positions across the court. We use a 40–10 split, predicting 10 frames of data from 40 given frames.

The first model is a naive setup where we attempt to predict trajectory data from the geometric data present in a frame alone. Using the Transformer encoder, the position data is fed as a sequence to just the encoder system (6 encoder layers) and a final linear layer to convert the Transformer memory into a concatenated vector of position data.

The second model is the full Transformer setup as described in Sect. 3.4 where the encoder accepts a full sequence of geometric data and the decoder operates on the geometric data as described.

Holding all factors constant, such as learning rate (0.0001), optimization algorithm (Adam), batch size (64), embedding dimension (200), number of layers (6), number of attention heads (8), and all other hyperparameters, we can see the effects of translating from the sequence of graphs through ablation.

The validation loss is set as the sum mean-square error of the 10 frames to be predicted. Namely, when predicting frames x_{41}, \dots, x_{50} with model output $\hat{x}_{41}, \dots, \hat{x}_{50}$, the validation error is

$$\frac{1}{10} \sum_{i=41}^{50} (x_i - \hat{x}_i)^2$$

Importantly, this experimental setup provides us with direct insight into how useful and fruitful the jump Markov model is. By using a naive baseline with Transformer, we essentially are performing an ablation analysis, where we see the effect of our entire graph extraction setup. In this context, the MSE score (i.e. the validation error defined above) provides us with a useful quantitative result on the utility of our entire pipeline.

Over the course of several training runs that were run to convergence, the naive model achieves an MSE score of approximately **280**, while the translation model achieves a best MSE score of **94** (lower MSE is better). This dramatic decrease in MSE through translation represents the effect of including the graph data, instead of attempting to learn position data directly.

5 Conclusions

This paper presents a semantic analysis of dynamic networks that are derived from geometric data. This idea is both novel in principle, but we demonstrate a quantitative benefit, as well, of using this approach. By converting geometric position data to graph data, we can leverage the gleaned structure to improve the overall accuracy of downstream applications, like trajectory prediction. Additionally, the construction of a jump Markov model provides clarity into the overall structure of a game through two important properties: first, a transition matrix that indicates which configuration of players lead to others and second, hold times, which suggest the stability of particular configurations in context. By constructing this jump Markov model, we can develop insights into our data that goes beyond the geometry.

From our experiments, we can conclude that, by incorporation the graph data, our pipeline does provide substantial practical insight. Through the completion-vs-translation setup, we can quantify exactly how much information is extracted out of this setup; the compelling drop in MSE garnered by the use of graph data underscores the importance of extracting and using the graph data.

Future work could use our semantic extraction toolkit on a variety of dynamic networks; dynamic networks appear in a plethora of natural contexts, and studying these sequences can be arduous. By analyzing them semantically, we can provide further insights into the dynamics. Additionally, though semantic extraction is a discipline that

is at least one hundred years old, we would like to build a more solid theoretical foundation for the particular case of semantic extraction from graphs.

In summary, this paper presents a novel model that leverages dynamic graphs and their semantics to provide a deeper understanding of geometric data. Basketball is but one source of geometric (or even network) data. The basic conceit of extracting semantics from data opens novel avenues of research that we intend to explore further in the future.

References

1. Baker, E.: Redefining Basketball Positions with Unsupervised Learning. *Towards data science* (2017)
2. Beckham, J.W.: Analytics Reveal 13 New Basketball Positions. *Wired* (2012)
3. Bialkowski, A., Park, S., Carr, G.P., Matthews, I., Yue, Y.: Analysis Of Team Behaviors Using Role And Formation Information (2018)
4. Cheng, A.: Using Machine Learning to Find the 8 Types of Players in the NBA. *Fastbreak Data* (2017)
5. Gonçalves, B., Coutinho, D., Santos, S., Lago-Penas, C., Jiménez, S., Sampaio, J.: Exploring team passing networks and player movement dynamics in youth association football. *PLoS ONE* **12**(1), 1–13 (2017)
6. Gould, P., Gatrell, A.: A structural analysis of a game: the Liverpool v Manchester united cup final of 1977. *Soc. Netw.* **2**(3), 253–273 (1979)
7. Gruber, C., Schwing, A.: Dynamic Neural Relational Inference for Forecasting Trajectories. *Cvpr* (2020)
8. Guðmundsson, J., Horton, M.: Spatio-temporal analysis of team sports. *ACM Comput. Surv.* **50**(2), 1–34 (2017)
9. Hobbs, J., Holbrook, M., Frank, N., Sha, L., Lucey, P.: Improved Structural Discovery and Representation Learning of Multi-Agent Data, pp. 1–16 (2019)
10. Li, M.G., Jiang, B., Zhu, H., Che, Z., Liu, Y.: Generative attention networks for multi-agent behavioral modeling. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI, pp. 7195–7202 (2020)
11. Lucey, P., Bialkowski, A., Carr, P., Morgan, S., Matthews, I., Sheikh, Y.: Representing and discovering adversarial team behaviors using player roles. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2706–2713 (2013)
12. Man, H.: Defining Modern NBA Player Positions - Applying Machine Learning to Uncover Functional Roles in Basketball. [Medium.com](#) (2017)
13. Morris, B.T., Trivedi, M.M.: A survey of vision-based trajectory learning and analysis for surveillance. *IEEE Trans. Circ. Syst. Video Technol.* **18**(8), 1114–1127 (2008)
14. Passos, P., Davids, K., Araújo, D., Paz, N., Minguéns, J., Mendes, J.: Networks as a novel tool for studying team ball sports as complex social systems. *J. Sci. Med. Sport* **14**(2), 170–176 (2011)
15. Rudenko, A., et al.: Human motion trajectory prediction: a survey. *Int. J. Robot. Res.* **39**(8), 895–935 (2020)
16. Schuette, C., Metzner, P.: Markov Chains and Jump Processes: An Introduction to Markov Chains and Jump Processes on Countable State Spaces. *Freie Universitaet Berlin* (2009)

17. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008. Curran Associates, Inc. (2017)
18. Yeh, R.A., Schwing, A.G., Huang, J., Murphy, K.: Diverse generation for multi-agent sports games. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4605–4614 (2019)
19. Yue, Y., Lucey, P., Carr, P., Bialkowski, A., Matthews, I.: Learning fine-grained spatial models for dynamic sports play prediction. In: *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 670–679 (2014)
20. Christopher, Z.: Redefining Positions and Players In Todays NBA. Using Machine Learning, *The Sports Scientist* (2019)



Graph-based Retrieval for Claim Verification over Cross-document Evidence

Misael Mongiovì^(✉) and Aldo Gangemi

ISTC - Consiglio Nazionale delle Ricerche, Catania and Rome, Italy
`{misael.mongiovi,aldo.gangemi}@istc.cnr.it`

Abstract. Verifying the veracity of claims requires reasoning over a large knowledge base, often in the form of corpora of trustworthy sources. A common approach consists in retrieving short portions of relevant text from the reference documents and giving them as input to a natural language inference module that determines whether the claim can be inferred or contradicted from them. This approach, however, struggles when multiple pieces of evidence need to be collected and combined from different documents, since the single documents are often barely related to the target claim and hence they are left out by the retrieval module. We conjecture that a graph-based approach can be beneficial to identify fragmented evidence. We tested this hypothesis by building, over the whole corpus, a large graph that interconnects text portions by means of mentioned entities and exploiting such a graph for identifying candidate sets of evidence from multiple sources. Our experiments show that leveraging on a graph structure is beneficial in identifying a reasonably small portion of passages related to a claim.

Keywords: Claim verification · Passage retrieval · Graph-based NLP

1 Introduction

Claim verification [1] is the task of deciding whether a claim is supported or refuted (or neither of them) by a reference knowledge base. It is a fundamental task in automated fact checking [2], with notable implications in the critical problem of contrasting disinformation, which has a significant worldwide impact [3]. Besides automatic fact checking, this task is relevant in every situation where the consistency of statements with respect to a knowledge base need to be verified. For instance, keeping a knowledge base updated when new information is available requires a consistency check of new data with previous knowledge. Integrating a supported claim would introduce redundancy, while a contradicted claim would make the knowledge base inconsistent. This is particularly important in robotics, in a scenario where a robot continuously updates its knowledge through conversation with humans. New statements that are supported by its internal knowledge might increase the confidence on encoded facts,

while statements that are refuted might indicate misunderstanding, unreliability of the interlocutor or the presence of false facts in the knowledge base.

Although modern language models are reasonably effective in deciding whether a short text (premise) infers or contradicts a claim (hypothesis), their performances fall when the size of the premise increases. Therefore, effective retrieval of related sentences from a reference corpus is crucial for accurate verification. Recent approaches on claim verification are indeed based on a retrieve-and-verify paradigm where given a claim, text passages related to the claim are first retrieved, then a classifier is employed to decide whether the retrieved text supports, refutes or is neutral about the claim. Thorne et al. [1] describe a general framework for verifying claims over a large corpus of text documents consisting in three steps. The *document retrieval* step extracts from the corpus a set of documents related to the claim, which are likely to contain the evidence for or against the claim. Then a *sentence selection* step performs a refinement by identifying in the retrieved documents a restricted set of sentences related to the claim, which we call the evidence. The last step, *claim verification*, decides whether the claim or its negation can be inferred by the selected set of sentences and classifies the claim into one of the labels *supported*, *refuted* or *not enough info*. Recently proposed claim verification methods are mostly based on this framework [4–11].

Although the described framework achieves significant accuracy on average, it struggles when multiple pieces of evidence, some of them little related to the claim, are distributed across different documents. As a toy example, the claim “The Beatles were formed in England” can be verified by the following two sentences: “The Beatles were formed in Liverpool” and “Liverpool is a city and metropolitan borough in Merseyside, England”. If we consider Wikipedia as the reference corpus, such sentences appear in separate documents, i.e. the page about “The Beatles” and the page about “Liverpool”. While “The Beatles” is clearly related to the claim, the relevance of “Liverpool” is negligible since it is not mentioned in the claim. Without other knowledge, its relevance would be no higher than any other city in England. Of course the Liverpool page can be identified by implicit or explicit background knowledge, but in general such knowledge is not always available, and even when it is, it might be considered unreliable. Some recent work (e.g. [12]) focus explicitly on the problem of retrieving documents related to a sentence (a claim or a question). Although they consider the distribution of related passages across documents, they do not give a specific solution and hence they still struggle when the evidence is fragmented into several document, each of them loosely related to the claim.

We propose a graph-based approach to retrieve relevant evidence for claim verification that translates the problem into a network search problem. The key idea is that the reference corpus can be summarized by a network that connects mentioned *entities* (named entities or, in general, entities that can be uniquely identified) that are referred contextually, whose exploration can help identifying relevant concepts and corresponding text portions. In the example above, the

path that connects the mentions to “The Beatles”, “Liverpool” and “England” in the reference corpus outlines the evidence for the claim.

In the remainder we first describe our graph-based approach for retrieving evidence across documents (Sect. 2), then we report the results of our experimental analysis (Sect. 3). We discuss related work (Sect. 4) and eventually conclude the paper and outline future work (Sect. 5).

2 Method

We generate an undirected multi-graph where nodes are *entities* mentioned in the reference corpus and two nodes are connected by one or more edges if the two entities are mentioned contextually. *Entity mentions* can be identified by entity linking tools (we employed BLINK [13]), which associate text spans referring to entities with corresponding entries in a knowledge base (in this work Wikipedia). To define when two entities are mentioned contextually, we introduce the concept of *frame*. In linguistics, a *semantic frame* [14] represents a set of concepts in a sentence which are related to a same action, event, or situation, usually described by a verb. In this paper, we generalize the concept of frame considering the possibility to be evoked by any text portion delimited by some rules. Therefore a *frame* can be evoked by a *sentence* or even a *paragraph* or a *document*, depending on the level of granularity we want to explore. We associate each edge of the graph to the location (document, sentence, text span) of the corresponding frame in the reference corpus.

Formally, we denote the multi-graph with $G = (V, E)$ where V is the set of vertices, composed by all entities mentioned in the reference corpus, and E is the set of edges, where an edge is a triple $e = (u, v, f)$ where $u, v \in V$ represent co-occurring entities and f is the reference to the frame where they occur. We also define $F(u, v)$ as the set of frames that contain mentions u and v in the reference corpus.

Given a claim c , we first extract all entity mentions from c by means of an entity linking tool [13]. We refer to the set of entities mentioned in c as M_c . The evidence can be traced by extracting from G a suitable subgraph that interconnects all entities in M_c through short paths (limited by some threshold l). A subgraph that interconnects entities in c outlines the set of frames (e.g., evoked by sentences) that might represent evidence for the claim. Specifically, the text associated to edges of the subgraph forms the candidate evidence for or against the claim. As a special case, when the claim contains just two entities u and v , all paths of length not above l from u to v (or, equivalently, from v to u) represent candidate evidence for the claim.

Figure 1 gives a general idea of the proposed method. In the center we show a fragment of the graph, where nodes are entities, and edges connect entities that are mentioned contextually. Edges are associated to the sentences where the entities co-occur in the reference corpus (pages in the bottom). The sentence in the top is the claim. It contains two named entities (“The Beatles” and “England”) that are disambiguated and connected to nodes in the graph. The

path that connects the two entities outlines the evidence for the claim, which is represented by the sentences associated to its edges.

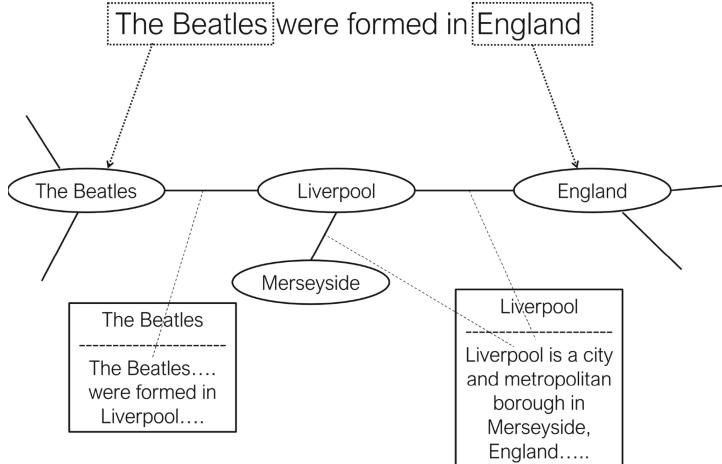


Fig. 1. An example of claim whose evidence is outlined by a path in the graph.

There might be a large number of different subgraphs that interconnect entities well. A possibility is to take the one that has the minimum number of edges. This lead to an optimization problem known as the *minimum Steiner tree* problem, which is NP-hard [15]. Besides its complexity, there is no guarantee that the minimum-size subgraph outlines the correct evidence for the claim. There are potentially many different minimal solutions and even a non-minimal variant might be a better fit. A different possibility is to enumerate all subgraphs that satisfy the connectivity constraints, but this solution would have exponential complexity. Instead we resort to a simple approach that focuses on selecting the part of the graph composed by edges and short paths between pairs of entities.

We limit l to be equal to 2 and consider three types of nodes: *mentioned nodes*, i.e. nodes of G that represent entities in M_c , *between nodes*, i.e. nodes that are not entities in M_c but are connected to at least two entities in M_c and all the remaining nodes, namely *unrelated nodes*. We remove unrelated nodes and all incident edges, and consider the remaining subgraph $G_c = (V_c, E_c)$ of G . We collect $\bigcup_{(u,v) \in E_c} F(u, v)$ as the candidate evidence for the claim c . We add to this set all sentences of pages that correspond to entities mentioned in the claim.

3 Experimental Analysis

We performed an experimental analysis aimed at assessing the effectiveness of the graph exploration in retrieving documents related to a claim. We considered

a subset of claims from the FEVER dataset [1]. FEVER contains 185K claims manually annotated with the information concerning whether they are supported or refuted (or neither of them) by a reference corpus made by 5.4M Wikipedia pages. For each supported or refuted claim FEVER provides all lines of evidence, consisting in all possible sets of sentences that support or refute the claim. Since our work is specifically focused on finding cross-document evidence, we selected claims whose evidence is across different pages. We also discarded claims that have less than two disambiguated entities and claims that contain entities that are too general (more than 1000 mentions in the corpus), obtaining a set of 2594 claims.

We implemented the proposed tool in Python 3.7. We employed the BLINK implementation from the authors¹ for entity linking. We considered sentences as frames. To manage the size of the graph (5.4M nodes and 68.8M edges), we stored it page by page in a key-value berkeleydb (version 5.3) database, where the key is the page ID and the value is the graph associated to the page. We also built an inverted index that associates entity mentions with the pages and the sentences they occur in. We applied the following procedure for each claim c : compute the set M_c of entities mentioned in c ; retrieve from berkeleydb all graphs of pages that mention entities in M_c ; get rid of edges of non-relevant sentences and combine the graphs; apply our tool described in Sect. 2.

We evaluated the proposed method, namely GraphRetrieve, against two baselines. The first, namely EntityRetrieve, returns all sentences of documents (Wikipedia pages) corresponding to disambiguated entities in the claim. The second, MentionRetrieve, return all sentences that mention at least one entity in the claim. We also combined the two baselines (Entity+MentionRetrieve) by merging sentences from both methods.

Results are reported in Table 1. We computed the “hit rate” as the percent of retrieval successes, i.e. the percent of claims for which all sentences of at least one line of evidence have been retrieved. This value has to be balanced with the amount of data retrieved, since a higher volume increases the chances to make a hit but would penalize the classification task. EntityRetrieve returns a small number of sentences (27.7 on the average, distributed across an average of 2.1 documents) but achieves the lowest hit rate (39.7%). Entity+MentionRetrieve achieves the highest hit rate (78.7%) but returns a significant amount of data (333.3 sentences over 250.5 documents, on average). GraphRetrieve achieves a fairly similar hit rate (70.1%) with a moderate volume of data (128.5 sentences), which is a little more than one third of the one retrieved by Entity+MentionRetrieve.

Computing entity linking on the 5.4M Wikipedia pages was an expensive offline task that required many days of parallel computation. The average running time for each claim (excluding the offline computation) is reported in the last column of Table 1². This time is dominated by the entity linking task on

¹ <https://github.com/facebookresearch/BLINK>.

² The computation was performed on a machine with 16 CPUs, 16 GB of RAM and a GPU NVIDIA Quadro P2200.

the claim (on average 4.22 s, not shown). As expected, the fastest method is EntityRetrieve, which performs in 4.24 s, while Entity+MentionRetrieve and GraphRetrieve are the slowest ones, performing in 5.04 and 5.47 s, respectively. The graph analysis costs less than 10% increase in running time.

Table 1. GraphRetrieve achieves a hit rate slightly lower than Entity+MentionRetrieve, with a little more than one third of retrieved sentences and a slight increase in computation time.

Method	Avg. sentences	Avg. documents	Hit rate	Avg. time(s)
EntityRetrieve	27.7	2.1	39.7%	4.24
MentionRetrieve	312.4	250.4	47.0%	4.26
Entity+MentionRetrieve	333.3	250.5	78.7%	5.04
GraphRetrieve (ours)	128.5	89.6	70.2%	5.47

We report in Table 2 three examples of claims whose evidence has been correctly retrieved by GraphRetrieve and MentionRetrieve, while EntityRetrieve was not able to correctly identify it. In all cases the evidence is not contained in the pages linked by the name entity disambiguator (BLINK). In the second sentence, the page “The Black Stallion (film)” is not considered because of an error of the disambiguator. The evidence can be found by retrieving all those sentences that mention at least one entity in the claim, but this produces a large volume of data in terms of number of sentences (column “mentioned entities”). Such volume of data can be significantly reduced by the graph analysis (last column). For instance, the claim “A singer in Got a Girl starred in Final Destination 3” is true because Mary Elizabeth Winstead is a singer in Got a Girl that has starred in the movie Final Destination 3. The evidence for such a claim is given by two of the 25 sentences that mention at least one of the entities “Got a Girl” and “Final Destination 3”. This number is reduced to 4 sentences by considering the subgraph spotted by GraphRetrieved, which is shown in Fig. 2. Edges are labeled with the page ID and the index of the sentence that mention its endpoints. Two of the edges (Got_a_Girl, 0 and Mary_Eliabeth_Winstead, 0) contain the evidence. A third edge (Wendy_Christensen, 1) refers to a sentence that states that Mary Elizabeth Winstead portrayed the main character Wendy Christensen in Final Destination 3 and hence can be considered as an alternative evidence for the claim. Three other edges connected with “United States” appear because Mary Elizabeth Winstead is stated to be “American” and the album “I Love You...”³ is also related to both United States and Got a Girl (it is “by American musical duo Got a Girl”).

Note that GraphRetrieve completely ignores the semantics of relations between entities. Despite its simplicity, it makes a fairly good job in identifying the correct evidence. It reduces considerably the volume of data retrieved, with a small cost in terms of relevant evidence loss.

³ “I Love You but I Must Drive Off This Cliff Now”.

Table 2. Three example of claims (the first three in the dataset) for which GraphRetrieve and MentionRetrieve find the correct evidence, in contrast with EntityRetrieve that is not able to do it. From left to right we report the claim, the evidence – as a list of pairs (page ID, sentence index) – and the number of sentences obtained by: entity pages, i.e. all those sentences of pages that correspond to entities; mentioned entities, i.e. all those sentences that mention at least one entity in the claim; graph edges, i.e. all those sentences outlined by the subgraph extracted as described in Sect. 2. The volume of data outlined by the graph analysis is always significantly smaller than the one by retrieving the sentences that mention entities of the claim.

Claim	Evidence	# Sentences from		
		Entity pages	Mentioned entities	Graph edges
A singer in Got a Girl starred in Final Destination 3	Mary Elizabeth Winstead, 0 Got a Girl, 0	26	25	4
Mickey Rooney was in a film based on the novel The Black Stallion by Walter Farley	The Black Stallion (film), 0 Mickey Rooney, 21	56	410	54
Emmy Rossum had a prominent role in a movie of which Maggie Greenwald was the director	Emmy rossum, 2 Songcatcher, 0	18	48	15

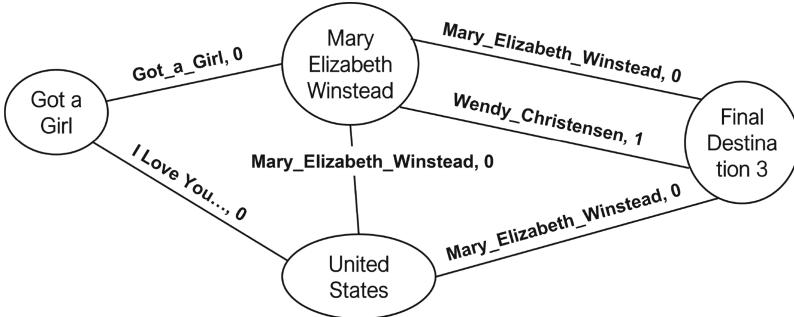


Fig. 2. The subgraph selected by GraphRetrieve for the claim “A singer in Got a Girl starred in Final Destination 3”.

4 Related Work

Claim verification is a fundamental step in fact-checking. Interested readers can find an extensive survey on the whole topic in [16]. Recent claim verification methods can rely on large annotated datasets to train machine learning models and achieve considerable results. Thorne et al. [1] provided FEVER, the first large-scale dataset with evidence for claim verification over a refer-

ence corpus consisting of 185,445 claims classified as *supported*, *refuted* or *not enough info* and associated to evidence from a corpus of 5.4 million Wikipedia pages. They described a pipeline that comprises the information retrieval and the textual entailment components. Recently proposed claim verification systems are mainly based on such a framework [4–11], where a retrieval component extracts sentences related to the claim from the corpus (the evidence), and a textual entailment component classifies the claim, based on the retrieved evidence. The retrieval component is usually decomposed in two sub-components: document retrieval, which identifies related documents, and sentence selection, which extracts salient sentences from the retrieved documents. The large size of FEVER enables training machine learning models for the task and obtaining performances that overcome 70% overall accuracy.

The document retrieval step is often shared among different works. A commonly used technique consists in retrieving a set of documents by keyword matching with the document titles [5,8] or calling the MediaWikiAPI⁴ of noun phrases from the claim [4,7,10,11]. Some methods also filter retrieved documents by a classifier based on NSMN [5,8,9], a variant of ESIM [17], a deep learning architecture based on two bidirectional LSTM (Long Short Term Memory) architectures. The claim and its noun phrases are compared with titles of previously retrieved documents to decide its relevance and filter out irrelevant documents. The sequence retrieval step is usually performed by a classifier that decides for every sentence of the retrieved documents whether it is related or not to the claim. Some systems employ ESIM [4,11], NSMN [8], or logistic regression [6] for this step. More recent systems employ transformers, the last generation language models, such as BERT [7,10] and XLNet [5].

A limit of the described approaches concerns the document retrieval phase. It gives no guarantee that available evidence for or against a claim is retrieved, since such evidence might be contained in documents whose titles might be loosely related or even not related at all to the claim. More advanced retrieval approaches [12,18–20] focus explicitly on the retrieval phase, aiming at selecting relevant content for diverse NLP tasks, including claim verification and question answering. They employ two encoders for embedding the documents and the query (e.g. a claim) into the same space, and perform a cosine similarity search to retrieve candidate documents. Eventually the search is refined by a cross-encoder classifier that combines each candidate document with the query and decides if it is relevant. The search can be performed at a finer level of granularity by considering short passages in place of complete documents. Although the described retrieval approaches have been proved successful in solving NLP tasks, including claim verification [12,20], they suffer when the evidence is fragmented across several documents, each of them loosely related to the claim. Our approach aims at overcoming this limit by interconnecting sentences of the reference corpus and providing a method for spotting all fragments of candidate evidence at once.

⁴ <https://www.mediawiki.org/wiki/API>.

5 Conclusion

We considered the problem of retrieving evidence for claim verification in the case when such evidence is distributed across different documents in a reference corpus. Available methods are not able to handle this case appropriately since each part of the evidence is retrieved independently. We interconnected all sentences in the reference corpus into a large graph and investigated whether the evidence can be identified in it as a subgraph. Despite the simplicity of the method, which do not even consider the semantics of the relation between entities, we are able to considerably reduce the amount of candidate evidence with a small loss of relevant text with respect to baseline approaches. As future work we plan to incorporate the semantics of relations between entities to make a more focused search and improve the method further.

Acknowledgment. The authors are partially supported by the Italian “Ministero dell’Università e della Ricerca” under the project “SI-ROBOTICS: SocIal ROBOTICS for active and healthy ageing” (PON 676 - Ricerca e Innovazione 2014-2020-G.A. ARS01_01120).

References

1. Thorne, J., Vlachos, A., Christodoulopoulos, C., Mittal, A.: FEVER: a large-scale dataset for fact extraction and VERification. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 809–819. New Orleans, Louisiana, Association for Computational Linguistics (June 2018)
2. Thorne, J., Vlachos, A.: Automated fact checking: Task formulations, methods and future directions. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 3346–3359 (2018)
3. Vosoughi, S., Roy, D., Aral, S.: The spread of true and false news online. *Science* **359**(6380), 1146–1151 (2018)
4. Zhou, J.: Gear: graph-based evidence aggregating and reasoning for fact verification. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 892–901 (2019)
5. Zhong, W.: Reasoning over semantic-level graph for fact checking. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6170–6180 (2020)
6. Yoneda, T., Mitchell, J., Welbl, J., Stenetorp, P., Riedel, S.: UCL machine reading group: four factor framework for fact finding (HexaF). In: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), pp. 97–102. Brussels, Belgium, Association for Computational Linguistics (November 2018)
7. Soleimani, A., Monz, C., Worring, M.: BERT for evidence retrieval and claim verification. *Adv. Inf. Retrieval* **12036**, 359–366 (2020)
8. Nie, Y., Chen, H., Bansal, M.: Combining fact extraction and verification with neural semantic matching networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 6859–6866 (July 2019). Number: 01

9. Ma, J., Gao, W., Joty, S., Wong, K.F.: Sentence-level evidence embedding for claim verification with hierarchical attention networks. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019), pp. 2561–2571 (August 2019)
10. Liu, Z., Xiong, C., Sun, M., Liu, Z.: Fine-grained fact verification with kernel graph attention network. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7342–7351 (2020)
11. Hanselowski, A.: UKP-athene: multi-sentence textual entailment for claim verification. In: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), pp. 103–108. Brussels, Belgium, Association for Computational Linguistics (November 2018)
12. Lewis, P., et al.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. [arXiv:2005.11401](https://arxiv.org/abs/2005.11401) (April 2021)
13. Wu, L., Petroni, F., Josifoski, M., Riedel, S., Zettlemoyer, L.: Scalable zero-shot entity linking with dense entity retrieval. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6397–6407 (2020)
14. Fillmore, C.J., Baker, C.F.: Frame semantics for text understanding. In: Proceedings of WordNet and Other Lexical Resources Workshop, NAACL, vol. 6 (2001)
15. Berman, P., Ramaiyer, V.: Improved approximations for the Steiner tree problem. *J. Algorithms* **17**(3), 381–408 (1994)
16. Guo, Z., Schlichtkrull, M., Vlachos, A.: A survey on automated fact-checking. [arXiv:2108.11896](https://arxiv.org/abs/2108.11896) (2021)
17. Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H., Inkpen, D.: Enhanced LSTM for natural language inference. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1657–1668 (2017)
18. Chang, W.C., Yu, F.X., Chang, Y.W., Yang, Y., Kumar, S.: Pre-training Tasks for Embedding-based Large-scale Retrieval. [arXiv:2002.03932](https://arxiv.org/abs/2002.03932) (February 2020)
19. Samarinis, C., Hsu, W., Lee, M.L.: Latent retrieval for large-scale fact-checking and question answering with NLI training. In: 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), pp. 941–948 (November 2020). ISSN: 2375–0197
20. Lee, K., Chang, M.-W., Toutanova, K.: Latent Retrieval for Weakly Supervised Open Domain Question Answering. [arXiv:1906.00300](https://arxiv.org/abs/1906.00300) (June 2019)



Supervised Link Weight Prediction Using Node Metadata

Larissa Mori¹, Mario Ventresca¹(✉), and Toyya A. Pujol²

¹ School of Industrial Engineering, Purdue University,
West Lafayette, IN 47906, USA
{lmori,mventresca}@purdue.edu

² Regenstrief Center for Healthcare Engineering, West Lafayette, IN 47906, USA
pujol@alum.mit.edu

Abstract. Given that node metadata can provide key insights about the relationship between nodes, we investigate if incorporating it as a similarity feature (referred to as *metadata similarity*) between end nodes of a link can improve the accuracy of weight prediction when using common supervised learning methods. We compare the weight prediction accuracy when metadata similarity is added to a set of baseline topological similarity features to that of using only the topological features. The comparison is performed across four empirical datasets using regression-based and other supervised methods found in the literature. In this preliminary study, we find no significant evidence that metadata similarity improves prediction accuracy in the methods analyzed and within the experimental setup. We encourage further investigation in this research area.

Keywords: Link weight prediction · Node metadata · Supervised machine learning

1 Introduction

Node metadata are observed discrete features or descriptors of nodes in a network [14]. For instance, authors' affiliation, country of origin, etc. would be considered node metadata in a network of co-authorships. Given that node metadata can provide key insights about the relationship between nodes [6], our intuition is that incorporating this information could improve link weight prediction accuracy. Thus, we investigate if incorporating node metadata as a similarity feature (referred to as *metadata similarity*) between end nodes of a link can improve the accuracy of weight prediction when using common supervised learning methods.

The use of node metadata is relatively sparse in the link prediction and link weight prediction literature. Zhao et al. posed link prediction as an optimization problem that minimizes the squared error regularized by some similarity measures, that can include metadata, if available [22]. A number of papers also approached the link prediction problem from a relational perspective. For example, Popescul and Ungar [15] built regression models including relations between

authors and venues into the prediction of co-authorships. Also, Taskar et al. [19] proposed a probabilistic relational model to incorporate relational data into the prediction of links.

There are a few studies that analyze the link weight prediction problem. Aicher et al. considered a stochastic block model to aid in weight prediction [2]. Zhao et al. calculated a linear relationship between weights and weighted versions of local similarity measures, inspired by the reliable route problem [21]. Assuming that edge weights are locally homogeneous, Zhu et al. proposed a metric that uses the weights of neighbor sets for prediction [24]. Fu et al. used supervised learning methods with a set of topological features extracted both from the original graph and the line graph, and also with a set of features derived with deep learning methods [7]. To the best of our knowledge, no previous work in the link weight prediction literature analyzed if the inclusion of similarity features based on node metadata improved link weight prediction accuracy.

Previous works in the link weight prediction literature focused on normalized weights to the interval $[0, 1]$, with the interpretation that link weights are “analogous to link-existence probabilities” [21]. However, we argue that link weights are better interpreted as a count variable indicating the number of interactions between two nodes. As a proxy for potential networks of interest, we note that most datasets [4, 8, 10, 20] that have been so far studied in the link weight prediction literature follow this interpretation. In these cases, prediction could suffer from the arbitrariness of the choice of normalization function and rounding errors incurred by mapping the original values to a $[0, 1]$ interval and then mapping the predicted ones back to integer values.

The remainder of this paper is structured as follows: Sect. 2 sets up the mathematical formulation of the problem and the metrics for accuracy comparison, Sect. 3 describes the methods used for prediction, Sect. 4 introduces metadata similarity and the set of baseline topological features used for comparison, Sect. 5 presents the analysis of real-world datasets and the experimental results, and finally Sect. 6 discusses the results and impact of the paper.

2 Problem Definition and Comparison Metrics

2.1 Problem Definition

Let $G = (V, E, W)$ be an undirected weighted network, where V denotes the set of nodes, E the set of links and W the set of link weights. Assume that there are no self-loops or multi-edges. In this study, we focus only on the problem of predicting the missing values of link weights in a network, assuming all links have been observed, i.e., when only information about W is incomplete.

For any link (i, j) between nodes i and j , let the corresponding weight $w_{ij} \in W$ be a positive integer sampled i.i.d. from a probability distribution $\mathcal{P}(w|\mathbf{x}, \boldsymbol{\theta})$, where \mathcal{P} is any probability distribution, $\mathbf{x} \in \mathbb{R}^p$ is the vector of edge features and $\boldsymbol{\theta} \in \mathbb{R}^q$ is a vector of model parameters. We would like to find an estimate $\hat{\boldsymbol{\theta}}$ for parameter values $\boldsymbol{\theta}$ that minimize the loss function: $\mathcal{L}(w, \hat{w}) = \mathbb{E}_{\mathbb{D}} [\text{dist}(w, f(\mathbf{x}, \boldsymbol{\theta}))]$, where the expectation is over the sample set of link weights \mathbb{D} , $\text{dist} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is any distance function, and $f : \mathbb{R}^p \times \mathbb{R}^q \rightarrow \mathbb{R}$ is a function determined by the user that estimates $\mathcal{P}(w|\mathbf{x}, \boldsymbol{\theta})$.

2.2 Comparison Metrics

Given that the actual distribution of link weights is unknown, we use the following procedure to test the prediction accuracy. First, we split the observed sample of link weights W into a training set W_{train} and a test set W_{test} , such that $W_{\text{train}} \cap W_{\text{test}} = \emptyset$ and $W_{\text{train}} \cup W_{\text{test}} = W$. In our experiments, we randomly sample 90% of the weights to compose W_{train} and the remaining 10% to compose W_{test} , as done previously in [7, 21, 24]. Then, we learn the parameters for each supervised learning method on W_{train} and calculate the accuracy of its predictions over W_{test} according to the set of metrics described below.

Previous weight prediction literature normalized the weight values to a $[0, 1]$ interval and then used the Root Mean Squared Error (RMSE) and the Pearson Correlation Coefficient (PCC) [21, 24]. However, since we are not normalizing the weights to a $[0, 1]$ interval, we use the Relative Squared Error (RSE), as it normalizes the error to indicate the percentage of improvement over “guessing the average” of the observed weights for each instance.

For notation simplicity and without loss of generality, we assume that the links in the observed sample are ordered and thus we collapse the indices i, j in w_{ij} to k , where $k = 1, \dots, N$, and N is the size of the test set W_{test} . In what follows, for every k , w_k is the observed weight value and \hat{w}_k is its predicted value.

- Pearson correlation coefficient (PCC):** It measures the linear correlation between two sets of data. It is a normalized measure of the covariance that falls between -1 and 1 , where values greater (smaller) than zero represent a positive (negative) correlation, and equal zero denote no correlation.

$$PCC = \frac{1}{N-1} \sum_{k=1}^N \left(\frac{w_k - \bar{w}}{\sigma_w} \right) \left(\frac{\hat{w}_k - \bar{\hat{w}}}{\sigma_{\hat{w}}} \right) \quad (1)$$

where $\bar{w} = \frac{\sum_{k=1}^N w_k}{N}$ is the average of the observed weights $w_k \in W_{\text{test}}$, and $\bar{\hat{w}} = \frac{\sum_{k=1}^N \hat{w}_k}{N}$ is the average of the predicted weights \hat{w}_k .

- Relative Squared Error (RSE):**

$$RSE = \frac{\left(\sum_{k=1}^N w_k - \hat{w}_k \right)^2}{\left(\sum_{k=1}^N w_k - \bar{w} \right)^2}. \quad (2)$$

3 Supervised Learning Methods

Given our assumption that link weights are positive integers representing a count variable, it is most natural to model them according to a Poisson distribution. Thus, we use Poisson regression as our baseline supervised learning method. We also compare the performance of this baseline with its extension to Poisson mixture models, allowing a weighted combination of different Poisson distributions. We note that the strong assumptions for the Poisson distribution (such that the mean and variance are equal) may not always hold for empirical datasets, thus

we leave for future work the comparison with other baseline methods for count variables, such as the negative binomial regression.

The main advantage of the regression-based methods are their statistical interpretability. One of the questions we would like to answer is whether, given our chosen set of features, regression-based methods can provide predictions that are at least as accurate as other commonly used methods in the literature. So, we also test their prediction accuracy against that of Random Forest and Support Vector Machine, that were used for weight prediction in a previous work [7].

In what follows, we detail the mathematical formulation of each model, according to the notation defined in Sect. 2.

Regression Methods

1. **Poisson regression (Poi):** The baseline estimator provides the least flexibility for estimation by constraining it to the functional form of a single Poisson distribution. Poisson regression is appropriate given that we assume that the weights are positive integers based on some frequency (e.g., of number of encounters, interactions, etc.). In this case, $\mathcal{P}(w|\mathbf{x}, \boldsymbol{\theta}) = \text{Poisson}(w|\mathbf{x}, \lambda(\mathbf{x})) = \frac{e^{-\lambda(\mathbf{x})}\lambda(\mathbf{x})^w}{w!}$, where $\lambda(\mathbf{x}) = \exp(\boldsymbol{\beta}^T \mathbf{x})$ and $\boldsymbol{\theta} = \boldsymbol{\beta}$.
2. **Poisson mixture models (Mix#):** This model generalizes our baseline estimator by accounting for subpopulations (components) within the broader population, each following a Poisson distribution with a different rate. In this study, we assume that the total number of components C is known. Formally, $\mathcal{P}(w|\mathbf{x}, \boldsymbol{\theta}) = \sum_{c=1}^C \pi_c \text{Poisson}(w|\mathbf{x}, \lambda_c(\mathbf{x}))$, where the mixing weights π_c satisfy $0 \leq \pi_c \leq 1$, $c = 1, \dots, C$, and $\sum_{c=1}^C \pi_c = 1$, $\lambda_c(\mathbf{x}) = \exp(\boldsymbol{\beta}_c^T \mathbf{x})$ and $\boldsymbol{\theta} = (\pi_1, \dots, \pi_C, \boldsymbol{\beta}_1^T, \dots, \boldsymbol{\beta}_C^T)$. We will refer to this method as Mix#, where # is the number of components assumed.

Methods for Comparison

3. **Random Forest (RF):** This method aggregates the decision criteria of many decision trees to avoid overfitting. It was the best performer overall among other unsupervised and supervised learning methods for weight prediction in a previous study [7], though with a different set of features. Formally, the decision tree model is described as: $\mathbb{E}[w|\mathbf{x}] = \sum_{m=1}^M h_m \mathbb{I}(\mathbf{x} \in R_m) = \sum_{m=1}^M h_m \phi(\mathbf{x}, \mathbf{y}_m)$, where h_m is the mean response in the m^{th} region R_m , and \mathbf{y}_m denotes the variable and its threshold value to split the tree into the m^{th} leaf. Thus, $\boldsymbol{\theta} = (\mathbf{y}_1, \dots, \mathbf{y}_M)$.
4. **Support Vector Machines (SVM):** This is the only method in this study that lacks a probabilistic interpretation. When used for regression, SVM consists of finding the line that best fits the data within an arbitrary boundary. Formally, this is done by solving the following minimization problem: $\underset{\boldsymbol{\beta}}{\text{minimize}} \frac{1}{2} \|\boldsymbol{\beta}\|^2 \text{ s.t. } |w_i - \boldsymbol{\beta}^T \mathbf{x}_i| \leq \epsilon, \forall i \in 1, \dots, |W_{\text{train}}|, \text{ for some } \epsilon > 0$.

For estimation of each of the methods, we use the R built-in function `glm` and packages `flexmix`, `randomForest` (with $n\text{tree} = 500$) and `e1071` (with the radial kernel and default hyperparameters), respectively.

4 Proposed Features

In this section we describe the features based on node metadata and network topology used for link weight prediction. We explain how we incorporate metadata about the nodes as a feature for link weight prediction and also describe the similarity features derived from network topology. In what follows, we denote the calculated similarity between nodes i and j according to a metric z by $s_{i,j}^z$, where the metric z can be based on node metadata or network topology. Then, the feature vectors used for prediction for an edge $k = (i, j)$ will be of the form $\mathbf{x}_k = [s_{i,j}^{z_1}, \dots, s_{i,j}^{z_p}]$, where each z_1, \dots, z_p is one of the similarity metrics considered.

4.1 Metadata Similarity Features: Incorporating Node Metadata

We assume that the metadata provides extra information about the nodes in the network. Since we would like to make predictions regarding the link weights, we need to incorporate the metadata available on the nodes as a feature about the links. There are many ways that this could be accomplished. In this study, we focus on calculating, for any link (i, j) , the similarity between its end nodes i and j , and using this result as one of the link's features in the supervised methods. We will refer to this feature as *metadata similarity*.

There are different metrics that can be used for the calculation of the similarity between two nodes i and j with respect to their metadata. Given a link (i, j) , let $\mathbf{v}_i, \mathbf{v}_j \in \mathbb{R}^t$ denote the vectors that store metadata information about end nodes i and j , respectively. In this study, we use the Pearson correlation coefficient and cosine similarity, as described below.

- Cosine similarity (CS):** This metric is defined as the cosine of the angle between two vectors. It can take values from -1 to 1 , where vectors with $CS = 1$ have the same orientation, $CS = -1$ are diametrically opposed, and $CS = 0$ are orthogonal to each other, regardless of their magnitude. It is defined as:

$$s_{i,j}^{CS} = \frac{\sum_{n=1}^t v_i^{(n)} v_j^{(n)}}{\sqrt{\sum_{n=1}^t v_i^{(n)2}} \sqrt{\sum_{n=1}^t v_j^{(n)2}}} \quad (3)$$

where t is the size of the metadata vector and $x_i^{(n)}$ is the n^{th} entry of \mathbf{x}_i .

- Pearson correlation coefficient (PCC):** For completeness, we restate its formulation as a similarity measure between end nodes:

$$s_{i,j}^{PCC} = \frac{1}{t-1} \sum_{n=1}^t \left(\frac{v_i^{(n)} - \bar{v}_i}{\sigma_{v_i}} \right) \left(\frac{v_j^{(n)} - \bar{v}_j}{\sigma_{v_j}} \right) \quad (4)$$

where \bar{v}_i and σ_{v_i} are the sample mean and standard deviation of \mathbf{v}_i , respectively (analogously for \mathbf{v}_j).

In the case where the metadata vector is comprised of different types of variables (numerical, ordinal and categorical), we calculate the similarity measures above on a new metadata vector where the categorical variables are transformed into dummy variables and the numerical/ordinal ones remain as before. For the measures considered, ordinal variables (such as “university rank”) do not need to be treated as dummies, as the difference between the “ranks” of two nodes is able to convey a notion of distance between them. This strategy has the limitation that it can only be applied when the total number of categories that the discrete variables take is small compared to the number of observations.

4.2 Topological Features

In order to analyze if there are prediction accuracy gains by adding metadata similarity to our prediction methods, we propose a set of topological features to be used as a baseline. There are many metrics proposed in the literature [7, 21]. In this paper, we focus on features that characterize the similarity between the end nodes of a link based on their local topology. Mainly, we consider metrics that take into account the number of neighbors in common between end nodes and/or their neighbors’ degrees. The list of the topological similarity features considered and their mathematical definition is described in Table 1.

Table 1. Local topological similarity measures used as baseline features.
Notation: $\Gamma(i)$ and k_i denote the set of neighbors and the degree of a node i , respectively, and $|\cdot|$ denotes the cardinality of a set.

Measure	Definition	Description
Common neighbors [12]	$s_{i,j}^{CN} = \Gamma(i) \cap \Gamma(j) $	Number of neighbors in common between i and j
Jaccard [9]	$s_{i,j}^{JI} = \frac{ \Gamma(i) \cap \Gamma(j) }{ \Gamma(i) \cup \Gamma(j) }$	The number of common neighbors divided by the number of nodes that are neighbors of at least one of the two nodes
Adamic Adar [1]	$s_{i,j}^{AA} = \sum_{z \in \Gamma(i) \cap \Gamma(j)} \frac{1}{\log k_z}$	The number of common neighbors of the vertices, weighted by the inverse logarithm of their degrees
Salton [17]	$s_{i,j}^{SA} = \frac{ \Gamma(i) \cap \Gamma(j) }{\sqrt{k_i \times k_j}}$	Cosine similarity. Measures the orientation similarity of the vectors
Resource allocation [23]	$s_{i,j}^{RA} = \sum_{z \in \Gamma(i) \cap \Gamma(j)} \frac{1}{k_z}$	The number of common neighbors of the vertices, weighted by their degrees
Sørensen [18]	$s_{i,j}^{SI} = \frac{2 \Gamma(i) \cap \Gamma(j) }{k_i + k_j}$	Twice the number of common neighbors divided by the sum of the degrees of the vertices
Hub depressed [16]	$s_{i,j}^{HDI} = \frac{ \Gamma(i) \cap \Gamma(j) }{\max(k_i, k_j)}$	With the denominator determined only by the higher degree, links adjacent to hubs are likely to be assigned low scores
Hub promoted [16]	$s_{i,j}^{HPI} = \frac{ \Gamma(i) \cap \Gamma(j) }{\min(k_i, k_j)}$	With the denominator determined only by the lower degree, links adjacent to hubs are likely to be assigned high scores
Leicht-Holme-Newman [11]	$s_{i,j}^{LHN} = \frac{ \Gamma(i) \cap \Gamma(j) }{k_i \times k_j}$	Node pairs with many common neighbors, as compared to the expected number of such neighbors, are assigned higher similarity
Preferential attachment [3]	$s_{i,j}^{PA} = k_i \times k_j$	Product of the two nodes’ degrees

5 Experimental Results

5.1 Data Description

For all datasets considered, the networks are undirected and weighted, and the link weights represent the number of interactions between its end nodes. Below we provide a description of the nodes and links in the networks analyzed and the metadata available in each case. In Table 2, we present some basic topological characteristics of the datasets.

1. *BusFac* [5]: Dataset of faculty transfers between universities. Nodes represent universities, edges between nodes indicate a transfer between one university to another (independent of direction), and weights the total number of transfers between two departments. Metadata available for each node: university's US News rank in 2012, Region (Northeast, Southeast, etc.), incoming/outgoing male percentage and incoming/outgoing assistant professor percentage.
2. *CompSciFac* [5]: Same node and edge structure of *BusFac*, where transfers refer to faculty in Computer Science departments.
3. *HistFac* [5]: Same node and edge structure of *BusFac* and *CompSciFac*, where transfers refer to faculty in History departments.
4. *BookCross* [25]: Subset of the dataset of the book review community Book Crossing. Nodes represent users, edges between two users indicate at least one book reviewed in common, and weights the total number of books reviewed in common. Metadata available for each node: country and age.

Table 2. Basic topological characteristics of datasets. $|V|$ and $|E|$ are the number of nodes and edges in the network, $\langle k \rangle$ is the average degree, $\langle S \rangle$ is the average link weight and r is the assortativity coefficient [13].

	$ V $	$ E $	$\langle k \rangle$	$\langle S \rangle$	r
<i>BusFac</i>	113	3515	62.212	2.572	-0.173
<i>CompSciFac</i>	206	2865	27.816	1.741	-0.109
<i>HistFac</i>	145	2334	32.193	1.944	-0.245
<i>BookCross</i>	240	26380	219.833	3.656	-0.050

5.2 Results

For each experiment, we display the average for each accuracy metric over 30 trials, where in each trial we randomly sampled 90% of the data for training and 10% for testing, as detailed in Sect. 2.2. For the prediction accuracy comparisons, we performed two-sided t-tests with significance of 5% (with the null hypothesis that the means of the sample errors are equal).

Comparison Between Methods. We compare the performance of the analyzed methods on the set of baseline topological similarity features. For the Poisson mixture models, we define the number of components in our experiments to be two to five components, and then compare their prediction accuracy to find the optimal number of components. It is worth noting, though, that this approach may not be feasible for large datasets and other methods for determining the optimal number of components may be more appropriate in those cases.

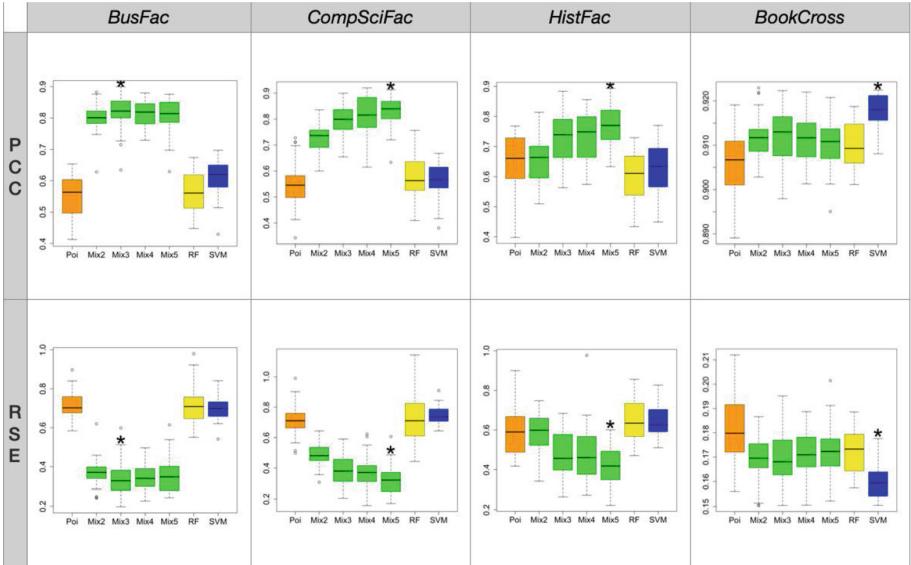


Fig. 1. PCC and RSE for methods with topological features. Each combination of row and column displays a boxplot comparison of the results for 30 iterations for each accuracy metric and dataset, where an asterisk is placed on top of the method with the best mean prediction accuracy. For *BusFac*, *CompSciFac*, *HistFac*, the methods with highest accuracy, for both metrics, were the mixture models with three, five and five components, respectively. For *BookCross*, SVM was the best performing method according to both metrics. For some datasets, mixture models can achieve the best performance when compared to other common supervised learning methods with the benefit of statistical interpretability.

The results are summarized in Fig. 1 for each of the datasets using the PCC and RSE metrics. For *BusFac*, the mixture model with three components (0.818 for PCC and 0.342 for RSE) outperformed, on average, the baseline Poisson regression (0.547 for PCC and 0.716 for RSE), RF (0.565 for PCC and 0.712 for RSE) and SVM (0.609 for PCC and 0.696 for RSE). For *CompSciFac*, the mixture model with five components (0.826 for PCC and 0.327 for RSE) outperformed, on average, the Poisson regression (0.554 for PCC and 0.705 for RSE), RF (0.574 for PCC and 0.724 for RSE) and SVM (0.609 for PCC and 0.696 for

RSE). Similarly, for *HistFac*, the mixture model with five components (0.774 for PCC and 0.415 for RSE) outperformed, on average, the Poisson regression (0.65 for PCC and 0.599 for RSE), RF (0.602 for PCC and 0.653 for RSE) and SVM (0.627 for PCC and 0.642 for RSE). For most cases, there was no significant statistical evidence to reject the null hypothesis that the mean of the sample errors is different when varying the number of components in the mixture models. For *BookCross*, the best performing method was the SVM (0.918 for PCC and 0.16 for RSE), but all methods (except for RF with 0.909 for PCC) performed, on average, better than the baseline Poisson regression (0.906 for PCC and 0.182 for RSE). For both metrics, the ranking of methods with respect to their prediction accuracy is consistent for all methods and datasets.

These results show that, for some datasets, mixture models can achieve the best performance when compared to other common supervised learning methods, given the set of topological similarity features used. For the *BookCross* dataset, the mixture model with three components had mean accuracy of 0.912 and 0.17 for PCC and RSE, respectively, whereas the SVM had an accuracy of 0.918 and 0.12. This might be sufficiently good for applications where accuracy improvements of the order of three or more decimal cases do not outweigh the benefit of statistical interpretability.

Comparison Using Metadata Similarity. As described in Sect. 4.1, we calculate metadata similarity features using both the cosine similarity and Pearson correlation coefficient measures. We test if the prediction accuracy of the analyzed methods improves when we include these metadata similarity features to the set of baseline topological features.

The results are displayed in Tables 3 and 4 for the regression-based methods and comparison methods, respectively. For each method X , we display side-by-side its average prediction accuracy for the set of topological plus metadata-based features and the set of topological features only (denoted by “m X ” and “t X ”, respectively), and indicate in bold the best performing one between the two sets. We also highlight in gray the pairs for which we reject the null hypothesis that the means are equal with statistical significance of 5%.

For the regression-based methods (Table 3), we cannot conclude that there are prediction accuracy improvements by adding metadata similarity features. There are some improvements for some methods and datasets, such as for Poisson regression on *BusFac* (from 0.58 to 0.55 for PCC, and 0.64 to 0.72 for RSE) and on *BookCross* (from 0.91 to 0.908 for PCC, and 0.182 to 0.179 for RSE), the prediction accuracy is actually lower for mixture models with three components for *CompSciFac* (from 0.775 to 0.758 for PCC, and 0.381 to 0.471 for RSE), for example.

For the comparison methods RF and SVM (Table 4), it is not possible to conclude either that there are accuracy improvements after the inclusion of metadata similarity. The results are mixed for all methods and datasets, where the prediction accuracy is higher for RF on *BusFac* (from 0.566 to 0.672 for PCC, and 0.712 to 0.568 for RSE) and *CompSciFac* (from 0.575 to 0.652 for PCC, and

Table 3. PCC (top) and RSE (bottom) comparison with metadata similarity features. Notation: $mPoi$ denotes the method Poi with both metadata and topological features and $tPoi$ the same method with only topological features (analogous for other methods), where the best performing one between the two is in bold. Also, the pairs for which we reject the null hypothesis that the means are equal are highlighted in gray, and in bold we highlight the best performing between the two. For most cases, the regressions with only the topological features performed better than when metadata similarity features are added to the baseline set, though most were not statistically significant.

	mPoi	tPoi	mMix2	tMix2	mMix3	tMix3	mMix4	tMix4	mMix5	tMix5
<i>BusFac</i>	0.5804	0.5469	0.7889	0.8004	0.8131	0.8176	0.8038	0.8126	0.807	0.8105
<i>CompSciFac</i>	0.5542	0.5397	0.7322	0.7727	0.7581	0.7749	0.803	0.8162	0.8179	0.8275
<i>HistFac</i>	0.6261	0.6358	0.6956	0.7168	0.7042	0.7623	0.759	0.7431	0.7449	0.7576
<i>BookCross</i>	0.9076	0.91	0.9106	0.912	0.9116	0.9121	0.913	0.9112	0.9126	0.9106
	mPoi	tPoi	mMix2	tMix2	mMix3	tMix3	mMix4	tMix4	mMix5	tMix5
<i>BusFac</i>	0.6434	0.7162	0.3954	0.3695	0.3901	0.3421	0.3732	0.3494	0.3598	0.3501
<i>CompSciFac</i>	0.7251	0.7049	0.4847	0.4906	0.4715	0.3812	0.3815	0.3724	0.3292	0.3273
<i>HistFac</i>	0.5944	0.5988	0.5362	0.5849	0.4601	0.4852	0.4767	0.4851	0.4385	0.4152
<i>BookCross</i>	0.1786	0.1817	0.1727	0.1694	0.1705	0.1703	0.168	0.1724	0.1691	0.1726

0.724 to 0.61 for RSE). On the other hand, the prediction accuracy is lower for SVM on *BookCross* with the inclusion of metadata features (from 0.918 to 0.916 for PCC, and 0.156 to 0.163 for RSE).

Table 4. PCC (left) and RSE (right) comparison for RF and SVM with metadata similarity features. Notation: mRF denotes the method RF with both metadata and topological features and tRF the same method with only topological features (analogous for other methods). For *BusFac* and *CompSciFac*, RF performed better when metadata similarity features were added to the baseline set. On the other hand, for *BookCross*, SVM performed worse with their inclusion, for both PCC and RSE. Overall, the results were mixed and most comparisons were not statistically significant.

	mRF	tRF	mSVM	tSVM		mRF	tRF	mSVM	tSVM
<i>BusFac</i>	0.6719	0.5655	0.6308	0.6086	<i>BusFac</i>	0.5677	0.7116	0.6395	0.6957
<i>CompSciFac</i>	0.6522	0.5745	0.5358	0.561	<i>CompSciFac</i>	0.6103	0.7239	0.7684	0.7461
<i>HistFac</i>	0.6128	0.6024	0.6611	0.6274	<i>HistFac</i>	0.6481	0.6526	0.6044	0.6419
<i>BookCross</i>	0.9091	0.9097	0.9158	0.9178	<i>BookCross</i>	0.174	0.173	0.1631	0.1597

It is worth noting that, in most cases, we failed to reject the null hypothesis that the mean performance given the different sets of features is the same. Given the limited availability of empirical weighted networks with node metadata, this study was limited to four empirical datasets, so further investigation is still required.

6 Conclusion

Though metadata is not widely available in empirical datasets yet, it is important to understand its role in weight prediction accuracy. As a stepping stone in this direction, we performed a set of experiments with empirical networks to understand if the inclusion of metadata similarity features to a set of baseline topological similarity features resulted in increased accuracy.

This preliminary study demonstrates that there are combinations of methods and networks where metadata does not improve prediction accuracy. Though the datasets analyzed may be small compared to many networks of interest nowadays, we note that these networks may share properties with and could represent community structures present in larger networks. However, a more comprehensive investigation with a larger range of methods and datasets is required to allow for more general conclusions.

In future work, we intend to expand this analysis to empirical networks with different characteristics (such as number of nodes, average weight, assortativity coefficient, etc.) and other supervised methods, such as the negative binomial regression discussed in Sect. 3. Also, as noted by Peel et al. [14] in the context of community detection, it is possible that the metadata available is irrelevant for network structure. Thus, it would be interesting to perform sensitivity analyses on synthesized networks and metadata to better understand under what conditions incorporating node metadata can help in weight prediction.

References

1. Adamic, L.A., Adar, E.: Friends and neighbors on the web. *Soc. Netw.* **25**(3), 211–230 (2003)
2. Aicher, C., Jacobs, A.Z., Clauset, A.: Learning latent block structure in weighted networks. *J. Complex Netw.* **3**(2), 221–248 (2015)
3. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* **286**(5439), 509–512 (1999)
4. Batagelj, V., Mrvar, A.: Pajek datasets. <https://vlado.fmf.uni-lj.si/pub/networks/data/> (2014). Accessed 18 July 2021
5. Clauset, A., Moore, C., Newman, M.E.J.: Hierarchical structure and the prediction of missing links in networks. *Nature* **453**(7191), 98–101 (2008)
6. Fajardo-Fontiveros, O., Sales-Pardo, M., Guimera, R.: Node metadata can produce predictability transitions in network inference problems. arXiv preprint [arXiv:2103.14424](https://arxiv.org/abs/2103.14424) (2021)
7. Fu, C., et al.: Link Weight Prediction Using Supervised Learning Methods and Its Application to Yelp Layered Network (2018)
8. Havugimana, P.C., et al.: A census of human soluble protein complexes. *Cell* **150**(5), 1068–1081 (2012)
9. Jaccard, P.: Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull. Soc. Vaudoise Sci. Nat.* **37**, 547–579 (1901)
10. Kunegis, J.: KONECT: the Koblenz network collection. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 1343–1350 (2013). <https://konect.uni-koblenz.de/>

11. Leicht, E.A., Holme, P., Newman, M.E.J.: Vertex similarity in networks. *Phys. Rev. E* **73**(2), 026120 (2006)
12. Newman, M.E.J.: Clustering and preferential attachment in growing networks. *Phys. Rev. E* **64**, 025102 (2001). <https://doi.org/10.1103/PhysRevE.64.025102>
13. Newman, M.E.J.: Assortative mixing in networks. *Phys. Rev. Lett.* **89**(20), 208701 (2002)
14. Peel, L., Larremore, D.B., Clauset, A.: The ground truth about metadata and community detection in networks. *Sci. Adv.* **3**(5), e1602548 (2017)
15. Popescul, A., Ungar, L.H.: Statistical Relational Learning for Link Prediction, p. 7 (2003)
16. Ravasz, E., Somera, A.L., Mongru, D.A., Oltvai, Z.N., Barabási, A.: Hierarchical organization of modularity in metabolic networks. *Science* **297**(5586), 1551–1555 (2002)
17. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
18. Sørensen, T.A.: A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biol. Skr.* **5**, 1–34 (1948)
19. Taskar, B., Wong, M.F., Abbeel, P., Koller, D.: Link prediction in relational data. *Adv. Neural Inf. Process. Syst.* **16**, 659–666 (2003)
20. Watts, D.J., Strogatz, S.H.: Collective dynamics of ‘small-world’ networks. *Nature* **393**(6684), 440–442 (1998)
21. Zhao, J., et al.: Prediction of links and weights in networks by reliable routes. *Sci. Rep.* **5**(1), 12261 (2015)
22. Zhao, Y., Wu, Y.J., Levina, E., Zhu, J.: Link Prediction for Partially Observed Networks (2017)
23. Zhou, T., Lü, L., Zhang, Y.C.: Predicting missing links via local information. *Eur. Phys. J. B* **71**(4), 623–630 (2009)
24. Zhu, B., Xia, Y., Zhang, X.J.: Weight prediction in complex networks based on neighbor set. *Sci. Rep.* **6**(1), 38080 (2016)
25. Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th International Conference on World Wide Web, pp. 22–32 (2005)

Graph Embedding



Joint Use of Node Attributes and Proximity for Node Classification

Arpit Merchant^(✉) and Michael Mathioudakis

University of Helsinki, Helsinki, Finland
{arpit.merchant,michael.mathioudakis}@helsinki.fi

Abstract. Node classification aims to infer unknown node labels from known labels and other node attributes. Standard approaches for this task assume homophily, whereby a node's label is predicted from the labels of other nodes nearby in the network. However, there are also cases of networks where labels are better predicted from the individual attributes of each node rather than the labels of nearby nodes. Ideally, node classification methods should flexibly adapt to a range of settings wherein unknown labels are predicted either from labels of nearby nodes, or individual node attributes, or partly both. In this paper, we propose a principled approach, JANE, based on a generative probabilistic model that jointly weighs the role of attributes and node proximity via embeddings in predicting labels. Experiments on multiple network datasets demonstrate that JANE exhibits the desired combination of versatility and competitive performance compared to baselines.

Keywords: Node classification · Graph embeddings · Spectral graph analysis

1 Introduction

Node classification involves a network with known graph structure, where each node is associated with a set of known attributes, as well as a label (or ‘class’) which is known only for some of the nodes and unknown for others. The task is to predict the unknown labels given all the known information about the network. This task finds application in many complex network settings such as information networks [21], complex systems [30], and protein function identification [5].

Previous Work. Motivated by theories of homophily [15] and social influence [14], a common assumption is that adjacent nodes tend to have similar labels. For instance, in a social network, friends may be likely to vote for the same political party. Methods for node classification that rely on this assumption typically enforce homophily by assigning the same label to proximate (i.e., nearby) nodes. In label propagation, for example, labels diffuse from labeled nodes to their unlabeled neighbors in an iterative manner until convergence [29]. Other methods induce label uniformity within cuts or clusters of the network [1, 2, 9];

or consider node proximity in a latent space that preserves network distances, as in DeepWalk [17] and similar matrix factorization methods [18].

However, the aforementioned methods ignore other node attributes, which can be detrimental. For example, Hamilton et al. [5] show that, for certain predictive tasks on citation and social networks, a linear classifier that is built only on node attributes outperforms methods such as DeepWalk that are based on node proximity but ignore node attributes. Similar cases are reported elsewhere in the literature [13, 20]. Therefore, it is important to appropriately leverage both the node proximity and node attributes, for label prediction.

Partially addressing this limitation, AANE [7] and DANE [4] combine low-dimensional encodings of node attributes with network-distance-preserving node embeddings, and use them as input for label prediction. However, they do not account for known labels during training, thus potentially ignoring information that would be useful in predicting the unknown labels. LANE [8] overcomes this limitation, by learning joint latent representations of node attributes, proximity, and labels. However, LANE does not directly address the node classification task, i.e., it does not optimize the conditional probability distribution of node labels given the node attributes and network structure, but rather targets their joint distribution of all quantities.

Separately, graph convolutional networks (e.g., GCN [26], GAT [23], GraphSAGE [5]) use network topology for low-pass filtering on node attributes. However, these convolutions are equivalent to repeated smoothing over the node attributes and performance quickly degrades [12]. Subsequent approaches such as DiffPool [28] have sought to address this limitation, but these too aggressively enforce homophily and require that nodes with the same labels have similar representations. Recent work by AM-GCN [25] attempts to weaken this assumption by analyzing the fusing capabilities of convolutional models. They define two modules, one each for the topology space and attribute space, and adaptively combine them using an attention mechanism.

Our Contribution. We develop an approach to node classification that flexibly adapts to a range of settings, from cases where node labels exhibit strong homophily (i.e., a node’s label can be predicted from the labels of proximate nodes) to cases where labels are solely predicted from attributes of the node, as well as cases that lie between these two extremes. Unlike aforementioned approaches that are heavily based on label homophily (e.g., label propagation), the proposed approach is based on a generative probabilistic model that jointly captures the role of network structure and node similarity in predicting labels. Our analysis leads to JANE, an algorithm that identifies a maximum-likelihood node labels. Moreover, unlike AANE [7] and DANE [4], JANE learns a low-dimensional node representation informed by labels, that is then used for prediction. And unlike LANE [8], it directly optimizes the conditional probability of unknown labels given the known information about the network. We empirically validate JANE’s performance on real datasets and compare to standard baselines.

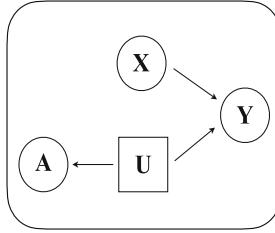


Fig. 1. The generative model. Observed node attributes \mathbf{X} (circular box) and latent attributes \mathbf{U} (square box) jointly generate node labels \mathbf{Y} . The (observed) adjacency matrix \mathbf{A} is generated from \mathbf{U} and indirectly correlates with \mathbf{Y} via \mathbf{U} .

2 Problem Setting

Let us consider a network with undirected and connected graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of node size $|\mathcal{V}| = n$. Let \mathbf{A} be the adjacency matrix, \mathbf{D} the (diagonal) degree matrix, and $\mathbf{L} = \mathbf{D} - \mathbf{A}$ the unnormalized Laplacian. Let $\mathbf{e}_i(\mathbf{L})$ be the i -th spectral eigenvector, corresponding to the i -th smallest eigenvalue of \mathbf{L} .

Each node j is associated with: d observed attributes $\mathbf{x}_j \in \mathbb{R}^d$, k latent attributes $\mathbf{u}_j \in \mathbb{R}^k$, and one (1) possibly unobserved label $\mathbf{y}_j \in \{1, 2, \dots, M\}$ as label. To refer to the attributes and labels of all nodes, we'll be using the notation $\mathbf{X} = \{\mathbf{x}_j\}_{j=1 \dots n}$, $\mathbf{U} = \{\mathbf{u}_j\}_{j=1 \dots n}$, and $\mathbf{Y} = \{\mathbf{y}_j\}_{j=1 \dots n}$, correspondingly.

Node classification is formalized as Problem 1.

Problem 1 (Node-Classification). Given adjacency matrix \mathbf{A} , node attributes \mathbf{X} , and labels \mathbf{Y}_L for a subset $L \subseteq \mathcal{V}$ of nodes, predict labels $\mathbf{Y}_{\mathcal{V}-L}$ for the remaining nodes $\mathcal{V} - L$ in the network.

3 Our Approach

Following a principled probabilistic approach for Problem 1 we make use of an appropriate generative model (defined in Sect. 3.1) to derive JANE, our node classification method (Sect. 3.2).

3.1 Model

Figure 1 illustrates our generative model. First, to model homophily, we assume that two nodes will be more likely to be adjacent in \mathbf{A} if they are similar in terms of some attributes. Formally, we assume that the adjacency matrix \mathbf{A} is generated from latent attributes \mathbf{U} . Specifically, the probability that there exists an edge between two nodes i and j is decreases with their distance in terms of \mathbf{u} according to the following formula, for some scaling parameter s^2 .

$$\Pr [(i, j) \in \mathcal{E} \mid \mathbf{u}_i, \mathbf{u}_j] = p_{ij} = e^{-\frac{\|\mathbf{u}_i - \mathbf{u}_j\|^2}{s^2}} \quad (1)$$

In light of the above, \mathbf{U} represents a low-dimensional Euclidean embedding of the nodes that preserves connectivity in the form of Eq. 1.

Second, to build the desired flexibility into our approach, we assume that labels \mathbf{Y} may be determined both from node attributes \mathbf{X} (directly) and node proximity as expressed directly via \mathbf{U} and thus indirectly via adjacency \mathbf{A} . Formally, we assume that this conditional probability of is given by a simple two-layer neural network,

$$\Pr[\mathbf{Y}|\mathbf{X}, \mathbf{U}, \mathbf{W}] = \sigma(\text{ReLU}([\mathbf{X}\mathbf{U}] W^{(0)})W^{(1)}) \quad (2)$$

where σ denotes the softmax function and weight matrices $\mathbf{W} = \{W^{(0)}, W^{(1)}\}$ are parameters that control the effect of \mathbf{X} and \mathbf{U} on labels \mathbf{Y} . We make this choice because we found this model to be sufficiently expressive for our empirical evaluation – however note that other models (e.g., neural networks with more hidden layers) could also be used in Eq. 2.

3.2 Algorithms

Problem 1 asks for predictions for $\mathbf{Y}_{\mathcal{V}-L}$ given the data $\mathcal{D} = (\mathbf{X}, \mathbf{A}, \mathbf{Y}_L)$ that are provided as input. Towards this end, we treat the remaining quantities in the model, i.e., the latent variables \mathbf{U} and the weights of the neural network \mathbf{W} , as unobserved parameters $\theta = (\mathbf{U}, \mathbf{W})$ – and use their maximum likelihood values $\hat{\mathbf{U}}$ and $\hat{\mathbf{W}}$ in making the predictions. In summary, JANE consists of two algorithms: first, a training algorithm from which we learn the maximum likelihood estimates $\hat{\mathbf{U}}$ and $\hat{\mathbf{W}}$; second, a prediction algorithm, in which we use the learned parameter values to predict the missing labels. We provide the algorithms below.

Training. The training algorithm is defined in Algorithm 1. In summary, it uses gradient descent on negative log-likelihood ($-\log \mathcal{L}$) to iteratively update $\hat{\mathbf{U}}$ and $\hat{\mathbf{W}}$ towards their maximum-likelihood estimates. The initial value of $\hat{\mathbf{U}}$ is given by the first spectral vectors of the network. Due to space constraints, the analysis that justifies this initialization and the detailed gradient calculations are omitted here but provided in the extended version [?].

Prediction. Given maximum-likelihood estimates $\hat{\mathbf{U}}$ and $\hat{\mathbf{W}}$, JANE invokes Eq. 2 with $\mathbf{U} = \hat{\mathbf{U}}$ and $W^{(=)}\hat{\mathbf{W}}$ for each node with unknown label, and predicts for it the most likely label according to Eq. 2.

4 Experiments

Baselines. We evaluate JANE against a range of baseline methods.

- **JANE and variants:** JANE-NU wherein we do not update the initial estimate of $\hat{\mathbf{U}}$ during training, and JANE-R where the initial estimate of \mathbf{U} is a random matrix.

Algorithm 1: JANE-Training

```

1 Input:  $\mathbf{A}; \mathbf{X}; \mathbf{Y}_L$ 
2 Output:  $\hat{\mathbf{U}} \hat{\mathbf{W}}$  (max-likelihood-estimates)
3 Parameters:  $k$  (dimensionality of  $\mathbf{U}$ );  $T$  (training epochs);  $\eta_1, \eta_2$  (learning rates)
    /* Initialize  $\hat{\mathbf{U}}$  with first spectral eigenvectors */
```

4 $\mathbf{L} \leftarrow \mathbf{D} - \mathbf{A}$

5 $\hat{\mathbf{U}} \leftarrow [\mathbf{e}_1(\mathbf{L}), \mathbf{e}_2(\mathbf{L}), \dots, \mathbf{e}_k(\mathbf{L})]$

for $t \leftarrow 1$ **to** T **do**

/* Update $\hat{\mathbf{W}}$ */

6 $\hat{\mathbf{W}} \leftarrow \text{backpropagation}([\mathbf{X} \hat{\mathbf{U}}], \hat{\mathbf{W}}, \mathbf{Y}_L)$

/* Update $\hat{\mathbf{U}}$ */

7 $\hat{\mathbf{U}} \leftarrow \hat{\mathbf{U}} - \eta_1 \cdot \frac{\partial C}{\partial \hat{\mathbf{U}}} - \eta_2 \cdot \frac{\partial (-\log \Pr[\mathbf{A} | \mathbf{U}])}{\partial \hat{\mathbf{U}}}$

end

- *Methods based on network structure:* Label Propagation (LP) [29], and DeepWalk (DW) [17] that encodes neighbourhood information via truncated random walks. These do not incorporate node attributes.
- *Deep attributed embeddings:* We evaluate LANE [8] which constructs node embeddings that encode network structure and node attribute information, in addition to node labels.
- *Methods based on graph-convolutions:* GCN [10] and GraphSAGE (mean aggregator) as representatives of graph convolutional networks. We acknowledge that this is extremely active area of research today and there are several methods that demonstrate improved performance along axes such as training efficiency [3], explainability [26], etc. on various real-world datasets [6]. For economy of space, we empirically compare with two benchmark, representative methods to demonstrate our central point: these methods sometimes fail because they strictly and inherently enforce homophily.

We were unable to reproduce the node classification results for AANE [7] based on the available implementation¹. Further, we could not locate the authors' implementations for DANE [4]. Therefore, we do not report results for them.

Experimental Setup. We implement JANE, JANE-NU, and LP in Pytorch. We use out-of-box Pytorch implementations of DeepWalk², GCN³ and GraphSAGE⁴. And, we use a MATLAB implementation⁵ of LANE [4]. In all of our experiments, as is standard, all methods receive only the adjacency matrix \mathbf{A} and the node attributes \mathbf{X} as input, along with the same 10% and 20% of the

¹ https://github.com/xhuang31/AANE_Python.

² <https://github.com/phanein/deepwalk>.

³ <https://github.com/tkipf/pygcn>.

⁴ <https://github.com/williamleif/graphsage-simple>.

⁵ <https://github.com/xhuang31/LANE>.

node labels for training and validation, respectively. Wherever available, we use the default hyperparameter configurations as suggested in the original papers. Otherwise, we grid search over the hyperparameter space to find the best setting for all of our baselines. We perform all experiments on a Linux machine with 4 cores and 32 GB RAM.

Reproducibility. To aid further research, we make our code publicly available⁶. This also includes an implementation for constructing synthetic datasets as described below. The real-world datasets (cf. Section 4.2) are publicly available.

4.1 Node Classification on Synthetic Data

The goal of these experiments is to demonstrate differences between JANE and existing classification methods.

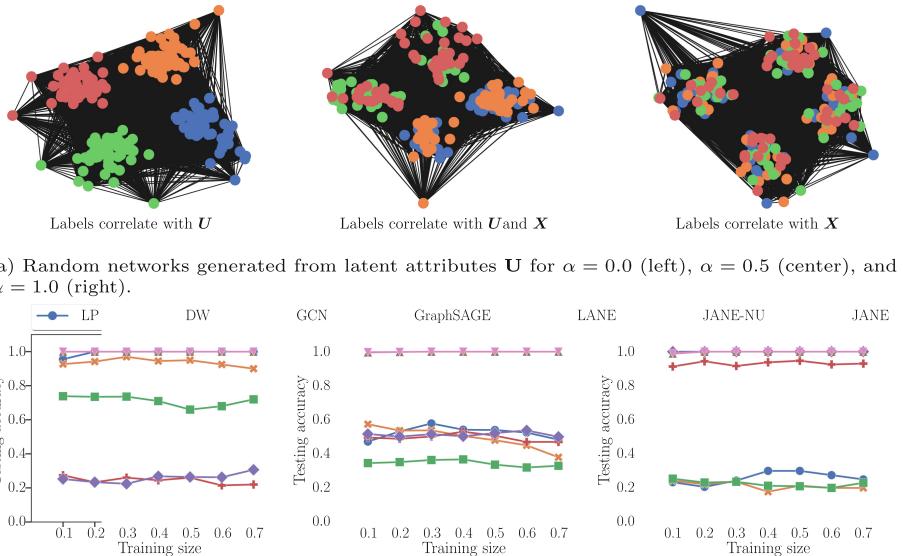
Synthetic Datasets. Figure 2 describes representative synthetic datasets generated according to the framework described previously. We set the number of individual node attributes $|\mathbf{X}| = d = 2$ and number of latent attributes $|\mathbf{U}| = k = 2$. We generate these attributes (gaussian-distributed) for $n = 200$ points, each of which belongs to one of $M = 4$ classes and set the scale $s^2 = 1$. An influence parameter, $\alpha \in [0, 1]$, controls the degree to which node labels derive from \mathbf{X} or \mathbf{U} : $\alpha = 0.0$ signifies that they derive only from \mathbf{U} and are independent of \mathbf{X} ; $\alpha = 1.0$ that they derive only from \mathbf{X} and are independent of \mathbf{U} ; and $\alpha = 0.5$ that they derive equally from \mathbf{X} and \mathbf{U} (specifically, without loss of generality, only the first attribute from \mathbf{X} and \mathbf{U} contributes to label assignment). Figure 2a depicts instances of synthetic networks corresponding to $\alpha = \{0.0, 0.5, 1.0\}$. The colors of points represent classes.

Implementation Details. We use Scikit-Learn’s [16] MAKE_CLASSIFICATION to generate these datasets. Classification algorithms do not have access to α or \mathbf{U} . JANE is trained as a two-layer neural network for a maximum of $T = 200$ epochs with dropout of 0.2 for each layer, weight decay of $5e^{-2}$, and learning rate of 0.005 using Adam. We set the number of eigenvectors $k = 2$ and choose a scaling factor $s^2 = 0.01$.

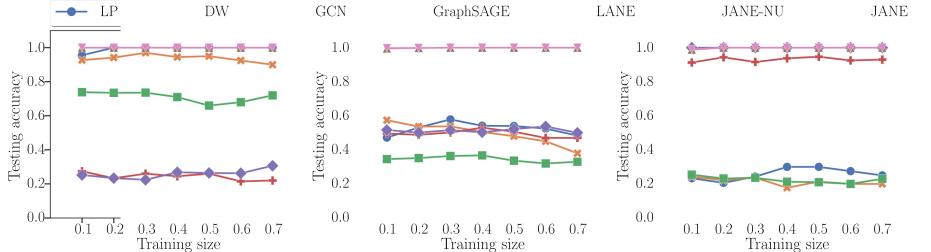
Performance. Figure 2b shows performance as a function of training set size.

- $\alpha = 0.0$: LP and DW infer that labels derive from \mathbf{A} (indirectly). GCN converges attribute values of nodes in the same cluster but is not perfectly accurate because \mathbf{X} does not correlate with \mathbf{Y} . LANE forces the proximity representation to be similar to the attribute representation and then smoothens it using the labels. It does not perform well since there is no correlation between them.
- $\alpha = 0.5$: LP, DW are able to correctly classify nodes belong to 2 out of 4 classes, i.e. precisely those nodes whose labels are influenced by \mathbf{U} . Conversely,

⁶ <https://version.helsinki.fi/ads/jane>.



(a) Random networks generated from latent attributes \mathbf{U} for $\alpha = 0.0$ (left), $\alpha = 0.5$ (center), and $\alpha = 1.0$ (right).



(b) Node classification accuracy averaged over randomly generated datasets for $\alpha = 0.0$ (left), $\alpha = 0.5$ (center), and $\alpha = 1.0$ (right).

Fig. 2. Figure 2a shows 3 synthetic networks where class labels \mathbf{Y} are determined only by \mathbf{U} ($\alpha = 0.0$), partly by \mathbf{U} and partly by \mathbf{X} ($\alpha = 0.5$), and only by \mathbf{X} ($\alpha = 1.0$). Figure 2b compares the node classification accuracy of JANE and JANE-NU with the baselines averaged over 5 random train-test splits.

LANE is able to classify those nodes belong to two classes of nodes that correlate with \mathbf{X} . GCN smoothes attribute values of adjacent nodes and thus can correctly infer labels correlated with \mathbf{X} .

- $\alpha = 1.0$ LP and DW reduce to random classifiers since adjacent nodes do not have similar labels. GCN reduces to a nearly random classifier because by forcing adjacent nodes with different attribute values to become similar, it destroys the correlation between \mathbf{X} and the labels.

In all cases, JANE-NU and JANE achieve perfect accuracy, as they flexibly learn whether labels are predicted from \mathbf{X} , \mathbf{A} (indirectly), or both. While these datasets are simplistic, this demonstrates how important this flexibility is.

4.2 Node Classification on Real-World Data

Datasets. We show our results on four tasks for a total of 9 networks (see Table 1 for basic statistics). If the original network is disconnected, we extract node attributes and labels belonging to its largest connected component.

- *Citation Networks:* We use Cora, Citeseer, Pubmed [21], and UAI2010 [24]. Here, nodes represent academic papers, edges denote a citation between two

Table 1. Summary of dataset statistics.

Dataset	Nodes	Edges	Classes	Attributes
Cora	2708	5429	7	1433
Citeseer	3327	4732	6	3703
Pubmed	19717	44325	3	500
UAI2020	3067	56622	18	4973
BlogCatalog	5196	171743	6	8189
Flickr	7575	239738	9	12047
Brazil	131	1038	4	NA
Europe	399	5995	4	NA
USA	1190	13599	4	NA
PPI	2373	56952	1	121

nodes, node attributes are 0/1-valued sparse bag-of-words vectors and class labels denote the subfield of research that the papers belong to.

- *Social Networks:* We focus on BlogCatalog and Flickr where the task is to predict pre-defined categories of blogs and images, respectively. Nodes are users that post content, edges represent follower relationships, and attributes are specified by a list of tags reflecting the interests of the users [11].
- *Air-traffic Networks:* Based on flight records from Brazil, Europe, and USA, each node is an airport and an edge indicates a commercial airline route exists between them. Labels denote the level of activity in terms of people and flights passing through an airport [19]. Since no attributes for the nodes exist, we assign the all-ones dummy vector as the sole attribute.
- *Biological Networks:* We use a processed protein-protein interaction (PPI) dataset [5] where the task is to identify protein roles based on gene ontology sets using positional gene sets, motif gene sets, and immunological signatures as attributes [22].

Experimental Setup. for the citation datasets, we use the same train-validation-test splits as in Yang, et al. [27] minus the nodes which do not belong to the largest connected component. These comprise of 20 samples for each class and represent 5% of the entire dataset. We use 500 additional samples as a validation set for hyperparameter optimization as per Kipf, et al. [10] to enable fair comparison. For all other tasks, we use 10% and 20% of the dataset for training and validation, respectively. We evaluate the performance of all methods on the remaining nodes. Values of hyperparameters k , the number of eigenvectors of the Laplacian, and the scaling factor s are determined so as to minimize the likelihood function (details omitted).

Performance Analysis. Table 2 provides the average accuracy of each method over 10 independent runs for $T = 200$ epochs each for GCN, GraphSAGE, LANE, and JANE. Values in bold denote that the method either performs best or its

Table 2. Classification accuracy (%) on test data averaged over 10 independent runs of $T = 200$ epochs each. Bold denotes best average accuracy or overlapping with best accuracy range (within ± 0.3 standard deviation). JANE-NU and JANE perform as well as or better than LANE, GCN and GraphSAGE on most datasets and consistently well across datasets.

Dataset	LP	DW	GCN	GraphSAGE	LANE	JANE-R	JANE-NU	JANE
Cora	74.29	32.54	79.20	79.70	64.50	59.34	78.87	79.24
Citeseer	67.19	61.31	69.11	69.84	56.93	52.80	69.71	69.77
Pubmed	64.33	77.91	82.26	81.86	78.81	72.91	82.70	83.18
UAI2010	42.18	44.67	49.52	61.22	66.36	45.96	66.91	69.84
BlogCatalog	48.81	37.18	70.53	75.13	82.26	70.85	74.85	77.53
Flickr	41.24	33.61	49.58	57.17	64.12	49.24	59.92	61.68
Brazil	26.83	51.71	27.21	28.26	24.59	5	29.32	68.55
Europe	26.33	47.15	25.46	46.61	27.50	28.81	45.97	50.21
USA	24.73	56.67	24.13	48.63	25.74	5	27.67	57.10
PPI	38.83	47.19	51.19	61.26	53.73	45.12	62.68	62.98

accuracy range overlaps with that of the best. JANE-NU and JANE consistently outperform LP and DW on all datasets by significant margins. JANE-R on the other hand, has significantly lower performance either because it gets stuck in a local optima or it requires much longer training time. This shows that the choice of the initial estimate is crucial to performance. The embedding obtained from DW is unsupervised and thus its predictive power is limited in comparison to our $\hat{\mathbf{U}}$ which is label-informed. Both variants of JANE are competitive with GCN and GraphSAGE on Cora, Citeseer, and UAI2020. For instance, GraphSAGE achieves 79.70% accuracy on Cora, while JANE gets 79.24% which is within the margin of error. We observe strong performance by JANE-NU and JANE on Pubmed and PPI. This may be explained in part by Li, et al. [12]’s observation that Pubmed exhibits strong manifold structure as well as JANE’s ability to better utilize the adjacency information. Lastly, we find significant gains on the social, and air-traffic datasets. LANE [8] report strong performance on Blogcatalog and Flickr, but its performance is noticeably poorer on citation, flights, and biological datasets.

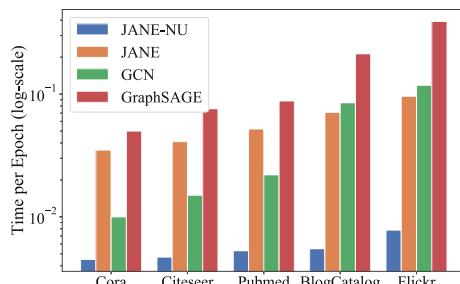


Fig. 3. Average training time per epoch (seconds) of baselines on different datasets.

Runtime. Figure 3 plots the average time for a single training epoch for JANE-NU, JANE, GCN, and GraphSAGE on various datasets. JANE is comparatively the slowest model while JANE-NU is by far the fastest. This is because JANE-NU is a vanilla neural network model that does not update its estimate of $\hat{\mathbf{U}}$. Further, a single laplacian eigenvector can be approximately computed using the Lanczos algorithm in $\tilde{\mathcal{O}}(|\mathcal{E}|)$ (up to log factors). Thus k eigenvectors can be computed in $\tilde{\mathcal{O}}(k \cdot |\mathcal{E}|)$ and this is a one-time operation. Note, we do not need to perform a full eigendecomposition.

Parameter Sensitivity. A crucial parameter of JANE is the number of latent attributes \mathbf{U} during training. Figure 4 demonstrates their impact on performance. In each case, JANE is allowed a maximum of 200 training epochs. We find that test accuracy consistently increases as dimension of $\hat{\mathbf{U}}$ increases up until a certain threshold. However, increasing beyond this threshold introduces noise and reduces performances. Having too many latent attributes gives marginal and diminishing returns in performance while increasing the runtime.

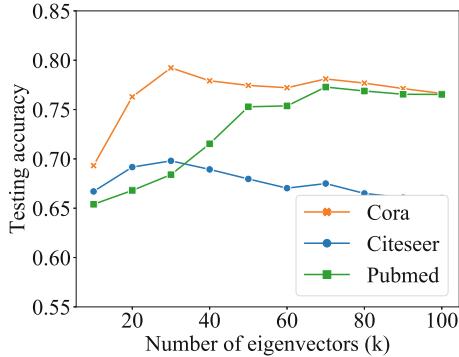


Fig. 4. JANE’s classification accuracy (%) on test data averaged over 10 runs w.r.t. number of eigenvectors.

Limitations. The primary limitation of JANE is the time and memory requirement for computing the gradient of \mathbf{A} w.r.t. $\hat{\mathbf{U}}$. These requirements grow linearly in network size. Since the gradient is computed in every training epoch, it may not be viable to fit it into GPU memory. Future work can outline procedures for mini-batch computation. However, note that JANE-NU is orders of magnitude faster than the other methods while also demonstrating strong performance.

5 Conclusion

We demonstrated an approach to node classification that flexibly adapts to settings where proximity and individual attributes have varying importance in predicting labels. Given its simplicity, interpretability and performance, JANE can

serve as a useful starting point in designing models that holistically account for different sources of node labels. As a future direction, we aim to evaluate the design and performance of advanced graph neural networks that go beyond requiring or enforcing homophily.

References

1. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.* **7**, 2399–2434 (2006)
2. Blum, A., Lafferty, J., Rwebangira, M.R., Reddy, R.: Semi-supervised learning using randomized mincuts. In: Proceedings of the Twenty-First International Conference on Machine Learning, p. 13. ACM (2004)
3. Cong, W., Forsati, R., Kandemir, M., Mahdavi, M.: Minimal variance sampling with provable guarantees for fast training of graph neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1393–1403 (2020)
4. Gao, H., Huang, H.: Deep attributed network embedding. In: IJCAI (2018)
5. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 1025–1035 (2017)
6. Hu, W., et al.: Open graph benchmark: Datasets for machine learning on graphs. arXiv preprint [arXiv:2005.00687](https://arxiv.org/abs/2005.00687) (2020)
7. Huang, X., Li, J., Hu, X.: Accelerated attributed network embedding. In: Proceedings of the 2017 SIAM International Conference on Data Mining, pp. 633–641. SIAM (2017)
8. Huang, X., Li, J., Hu, X.: Label informed attributed network embedding. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, pp. 731–739 (2017)
9. Joachims, T.: Transductive learning via spectral graph partitioning. In: Proceedings of the 20th International Conference on Machine Learning (ICML-2003), pp. 290–297 (2003)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR) (2017)
11. Li, J., Hu, X., Tang, J., Liu, H.: Unsupervised streaming feature selection in social media. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management (2015)
12. Li, Q., Han, Z., Wu, X.M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
13. Lorrain, F., White, H.C.: Structural equivalence of individuals in social networks. *J. Math. Soc.* **1**(1), 49–80 (1971)
14. Marsden, P.V., Friedkin, N.E.: Network studies of social influence. *Sociol. Methods Res.* **22**(1), 127–151 (1993)
15. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: homophily in social networks. *Ann. Rev. Soc.* **27**(1), 415–444 (2001)
16. Pedregosa, F., et al.: Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)

17. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710 (2014)
18. Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., Tang, J.: Network embedding as matrix factorization: unifying deepwalk, Line, PTE, and node2vec. In: Proceedings of the eleventh ACM International Conference on Web Search and Data Mining, pp. 459–467 (2018)
19. Ribeiro, L.F., Saverese, P.H., Figueiredo, D.R.: struc2vec: learning node representations from structural identity. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 385–394. ACM (2017)
20. Sailer, L.D.: Structural equivalence: meaning and definition, computation and application. *Soc. Netw.* **1**(1), 73–90 (1978)
21. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. *AI Mag.* **29**(3), 93–93 (2008)
22. Subramanian, A., et al.: Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci.* **102**(43), 15545–15550 (2005)
23. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint [arXiv:1710.10903](https://arxiv.org/abs/1710.10903) (2017)
24. Wang, W., Liu, X., Jiao, P., Chen, X., Jin, D.: A unified weakly supervised framework for community detection and semantic matching. In: Phung, D., Tseng, V.S., Webb, G.I., Ho, B., Ganji, M., Rashidi, L. (eds.) PAKDD 2018. LNCS (LNAI), vol. 10939, pp. 218–230. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93040-4_18
25. Wang, X., Zhu, M., Bo, D., Cui, P., Shi, C., Pei, J.: AM-GCN: adaptive multi-channel graph convolutional networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1243–1253 (2020)
26. Wu, J., He, J., Xu, J.: Net: Degree-specific graph neural networks for node and graph classification. arXiv preprint [arXiv:1906.02319](https://arxiv.org/abs/1906.02319) (2019)
27. Yang, Z., Cohen, W., Salakhudinov, R.: Revisiting semi-supervised learning with graph embeddings. In: International Conference on Machine Learning, pp. 40–48. PMLR (2016)
28. Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., Leskovec, J.: Hierarchical graph representation learning with differentiable pooling. In: NeurIPS, pp. 4800–4810 (2018)
29. Zhou, D., Huang, J., Schölkopf, B.: Learning from labeled and unlabeled data on a directed graph. In: Proceedings of the 22nd international conference on Machine learning - ICML 2005, pp. 1036–1043. ACM Press, Bonn, Germany (2005). <http://portal.acm.org/citation.cfm?doid=1102351.1102482>
30. Zhu, X., Goldberg, A.B.: Introduction to semi-supervised learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **3**(1), 1–130 (2009)



Online Updates of Knowledge Graph Embedding

Luo Fei¹, Tianxing Wu², and Arijit Khan^{1(✉)}

¹ Nanyang Technological University, Singapore, Singapore

{fei.luo,arijit.khan}@ntu.edu.sg

² Southeast University, Nanjing, China

tianxingwu@seu.edu.cn

Abstract. Complex networks can be modeled as knowledge graphs (KGs) with nodes and edges denoting entities and relations among those entities, respectively. A knowledge graph embedding assigns to each node and edge in a KG a low-dimensional semantic vector such that the original structure and relations in the KG are approximately preserved in these learned semantic vectors. KG embeddings support downstream applications such as KG completion, classification, entity resolution, link prediction, question answering, and recommendation. In the real world, KGs are dynamic and evolve over time. State-of-the-art KG embedding models deal with static KGs. To support dynamic updates (even local), they must be retrained on the whole KG from scratch, which is inefficient. To this end, we propose a new context-aware Online Updates of Knowledge Graph Embedding (OUKE) method, which supports embedding updates in an online manner. OUKE learns two different vectors for each node and edge, i.e., knowledge embedding and context embedding. This strategy effectively limits the impacts of a local update in a smaller region, so that OUKE is able to efficiently update the KG embedding. Experiments on the link prediction in dynamic KGs demonstrate both effectiveness and efficiency of our solution.

Keywords: Knowledge graphs · Embedding · Dynamic updates

1 Introduction

Knowledge graph is a data model for complex networks to manage large-scale and real-world facts [9, 14]. Examples include DBpedia [18], YAGO [13], Freebase [4], NELL [22], personalized health knowledge graphs [11], etc., where a node represents an entity, and an edge denotes a relationship between two entities. Knowledge graph embedding [3, 32] is increasingly becoming popular, which aims to represent each relation and entity in a knowledge graph \mathcal{G} as a d -dimensional vector, such that the original structure and relations in \mathcal{G} are approximately preserved in this semantic space. KG embeddings are used in downstream applications, e.g., link prediction [27, 31, 36], entity classification [37], question answering [14, 33], KG completion [6], and recommender systems [38].

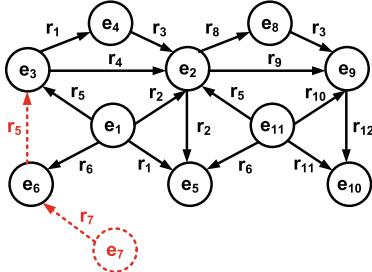


Fig. 1. Dynamic updates in a KG: dashed nodes and edges denote updates.

In the real world, KGs are dynamic and evolve over time [19, 26]. DBpedia extracts the update stream of Wikipedia each day to keep the KG up-to-date [12]. IMDB provides daily dumps of movies, TV series, actors, directors, among others, as well as their relationships [1]. Amazon product KG is updated quite frequently because there is a large number of new products everyday [9]. Recently, IBM’s COVID-19 knowledge graph can ingest 100 000 PDF pages per day [2]. However, existing models [3, 5, 7, 10, 24, 30, 34, 36] embed static KGs. To support dynamic updates in a KG, these models must be retrained on the whole KG from scratch, which is inefficient, and is also impracticable when the KG has higher update frequency (e.g., once per day). To this end, we study the novel problem of efficiently updating a KG embedding in an online manner.

Consider the KG in Fig. 1, the updates are denoted by dashed nodes and edges. Assume that we have embedded this KG using TransE [5], which should hold the translation relation: $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ (vectors are represented as bold characters) for each triple (h, r, t) , where h is a head entity, r is a relation, and t is a tail entity. For instance, the triple $(Leonardo\ da\ Vinci, creator, Mona\ Lisa)$ denotes that *Leonardo da Vinci* is the *creator* of the *Mona Lisa*. After adding a new triple (e_6, r_5, e_3) , where e_6, e_3 are existing entities and r_5 is an existing relation in the earlier version of the KG, we now need to satisfy $e_6 + r_5 \approx e_5$. No matter for which element in (e_6, r_5, e_3) we decide to update its vector, it will break the translation relations $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ for other triples containing our selected element, thereby creating a cascade of updates on the embedding of the entire KG. In summary, when a KG has local updates with addition and deletion of triples, if we revise the vectors of some entities and relations due to such updates, these revisions may cascade in the entire KG via connections among entities and relations, which is expensive.

When local updates occur in a knowledge graph, in the context of KG embedding, can we limit the impacts of such updates in certain regions, and not in the entire KG? To this end, we design a novel, context-aware Online Updates of Knowledge Graph Embedding (OUKE) approach, that answers this question affirmatively. We assign two different vectors to each entity and relation. When an entity (or a relation) denotes itself, we use a vector, called the *knowledge embedding*. When it denotes a part of the context of other entities (or relations),

we use another vector, referred to as the *contextual element embedding*. In the neighborhood of an entity (or a relation), contextual element embeddings are aggregated to form the *contextual subgraph embedding* via a Relational Graph Convolution Network (R-GCN) [25]. We construct the *joint embedding* of each entity (\mathbf{h}^* or \mathbf{t}^*) and relation (\mathbf{r}^*) by combining the knowledge embedding (i.e., \mathbf{h}^k , \mathbf{t}^k , or \mathbf{r}^k) and the contextual subgraph embedding (i.e., $\mathbf{sg}(h)$, $\mathbf{sg}(t)$, or $\mathbf{sg}(r)$) via a gate strategy. Finally, we employ the joint embedding of each entity or relation to hold the translation relation: $\mathbf{h}^* + \mathbf{r}^* \approx \mathbf{t}^*$.

We next propose an online learning algorithm to incrementally update the KG embedding. **(1)** Following the inductive learning, we keep all learnt parameters in R-GCNs and the gate strategy unaffected. **(2)** Contextual element embeddings of existing entities and relations also remain the same. **(3)** After a KG update, for many entities and relations, their contexts remain unchanged, so their contextual subgraph embeddings would remain uninterrupted. Thus, with existing knowledge embeddings of such entities and relations, corresponding triples would satisfy: $\mathbf{h}^* + \mathbf{r}^* \approx \mathbf{t}^*$. Hence, we also keep the knowledge embeddings of existing entities and relations unchanged so long as their contexts are unchanged. **(4)** What shall we do with an existing entity or relation having changed context? Notice that its contextual subgraph embedding, a combination of context element embeddings of its neighboring entities or relations, computed by the R-GCN, will change to reflect this update. We next relearn the knowledge embeddings of existing entities and relations with changed contexts, and in that process we adjust both their knowledge embeddings and joint embeddings, with the aim that the joint embeddings, after such update, still approximately satisfy the translations in the modified graph. **(5)** In addition, we also learn knowledge embeddings and contextual element embeddings of emerging entities and relations. In this way, our algorithm greatly reduces the number of triples which need to be retrained while preserving $\mathbf{h}^* + \mathbf{r}^* \approx \mathbf{t}^*$ on the whole KG. This enables online learning with higher efficiency.

Example 1. In Fig. 1, after adding triples (e_7, r_7, e_6) and (e_6, r_5, e_3) into KG \mathcal{G} , we have an emerging entity e_7 , an emerging relation r_7 , four existing relations with changed contexts r_1, r_4, r_5, r_6 , and two existing entities with changed contexts e_3 and e_6 . Based on our online learning, we retrain only nine triples having $e_3, e_6, e_7, r_1, r_4, r_5, r_6$, and r_7 ; i.e., (e_3, r_1, e_4) , (e_3, r_4, e_2) , (e_1, r_5, e_3) , (e_1, r_6, e_6) , (e_6, r_5, e_3) , (e_1, r_1, e_5) , (e_7, r_7, e_6) , (e_{11}, r_5, e_2) , and (e_{11}, r_6, e_5) and not all eighteen triples in the updated version of \mathcal{G} . This demonstrates the efficiency improvement due to our method even with such a small KG. In particular, we learn knowledge embeddings, contextual element embeddings, and joint embeddings of the emerging entity e_7 and the emerging relation r_7 . For existing relations with changed contexts (r_1, r_4, r_5, r_6) and existing entities with changed contexts (e_3 and e_6), their contextual element embeddings remain the same, but the contextual subgraph embeddings are updated via R-GCN, due to changed contexts. We also learn their updated knowledge embeddings and joint embeddings.

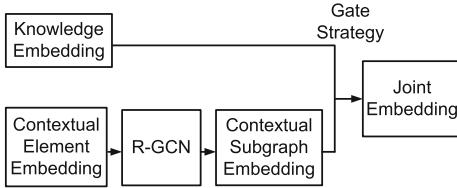


Fig. 2. Architecture of OUKE: learning from scratch.

We empirically evaluate our method, OUKE (Online Updates of Knowledge Graph Embedding) on link prediction over dynamic KGs. Compared to static KG embedding models, OUKE has comparable effectiveness in different evaluation metrics, and better efficiency in online learning since the static models must be retrained on the entire KG.

Related Work. Static KG embedding can be translation-based (e.g., TransE [5], TransH [34], and TransR [20]), via compositions of head-tail entity pairs with their relations (e.g., RESCAL [24], DisMult [36], ComplEx [30]), as well as neural network-based (e.g., R-GCN [25] and ConvE [7]). GAKE [10] simultaneously models triples and structural contexts in embedding learning on static KGs.

Temporal KG embedding has been considered in [8, 21, 28] over multiple given snapshots of a KG, by incorporating time in the entity-relation space, to better perform link prediction, time prediction, and future fact prediction. In other words, they only conduct offline embedding learning with multiple given KG snapshots, but when faced with KG updates, they also need to be retrained on the whole KG, so unlike ours, they cannot embed dynamic KGs with high efficiency. To the best of our knowledge, puTransE [27] is the only existing model supporting online embedding learning for dynamic KGs. puTransE learns embeddings of entities and relations from local parts of a KG, so it avoids retraining on the entire KG, but cannot preserve the global structural information of the KG in the learnt embedding. When compared with puTransE [27], our proposed method, OUKE generally outperforms it in both effectiveness and efficiency.

Several dynamic graph embedding methods are also developed, e.g., [29, 39]. They incrementally compute the embeddings of new nodes and update existing node embeddings after a graph update. However, when we need to learn edge (i.e., relation) embeddings and consider various semantic correlations among nodes and edges in a dynamic KG embedding, these models cannot be applied.

2 Problem Formulation

We categorize the problem of learning KG embedding in a dynamic scenario as two sub-problems: Learning from scratch and online learning. Let a KG $\mathcal{G}^T = \{(h, r, t)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where $\{(h, r, t)\}$ denotes a set of triples, h is a head entity, r is a relation, t is a tail entity, e.g., $\{BMW, product, Germany\}$ is a triple in a KG indicating that *BMW* is *produced* in *Germany*, \mathcal{E} and \mathcal{R} are the sets

of all entities and relations in \mathcal{G} , respectively, and \mathcal{T} is the current time step. We assume that each entity e in a KG has at least one type [23, 33], denoted by $T(e)$, e.g., *BMW* is of type *Automobiles* and *Germany* is of type *Country*.

Problem 1. Learning from Scratch. KG $\mathcal{G}^{\mathcal{T}}$ is given as an input to our method at time step \mathcal{T} . The outputs are vector representations of entities and relations. Each entity or relation has two vectors: knowledge embedding and context element embedding.

At time step $\mathcal{T} + 1$, $\mathcal{G}^{\mathcal{T}}$ becomes $\mathcal{G}^{\mathcal{T}+1}$ with updates including addition and deletion of triples, even having emerging entities and relations. We define *online learning* as follows:

Problem 2. Online Learning. The inputs are KG $\mathcal{G}^{\mathcal{T}+1}$ at time step $\mathcal{T} + 1$, KG $\mathcal{G}^{\mathcal{T}}$ at time step \mathcal{T} , and earlier embedding at time step \mathcal{T} . Our method OUBE outputs updated and new vectors for entities and relations at time step $\mathcal{T} + 1$.

3 Embedding Learning from Scratch

The architecture of learning from scratch in OUBE is presented in Fig. 2. We assign two different vectors to each entity or a relation: *knowledge embedding* and *contextual element embedding*.

Learning from scratch involves two phases: (1) **context encoding** models the context of each entity or relation as a (multi)graph, and then utilizes a relational graph convolutional network (R-GCN) [25] to encode such contexts; (2) **embedding learning** applies a gate strategy to aggregate knowledge and context vectors, and then defines a loss function based on translation for training.

Context Encoding. We define the context of each entity as an undirected subgraph consisting of its neighbor entities (connected via some relations) and itself. For efficiency of OUBE, we only consider one-hop neighbor entities. In our experiments, when we consider more distant neighbors besides one-hop ones, it consumes much more time for model training, but OUBE’s accuracy in link prediction does not significantly improve. This is because the further away entities are from each other, the less relevance they have [15–17], and less relevant neighbors may also introduce noise, in addition to useful information. Considering these trade-offs and based on our experimental results, we choose one-hop neighbor entities to build the context of each entity.

Different from entities, each relation occurs many times in a KG. We define the context of each relation as an undirected multi-graph consisting of all its neighbor relations (connected via entities having different types) and itself.

Example 2. In Fig. 3, we show the contexts of entity e_4 and relation r_2 , respectively, from the example KG in Fig. 1. The context of e_4 consists of itself, and its neighboring entities, e_3 and e_2 , connected via relations r_1 and r_3 , respectively. So, the context of an entity forms a subgraph of the input KG. The context of r_2 consists of itself and other neighboring relations: $r_1, r_3, r_4, r_5, r_6, r_8$, and r_9 .

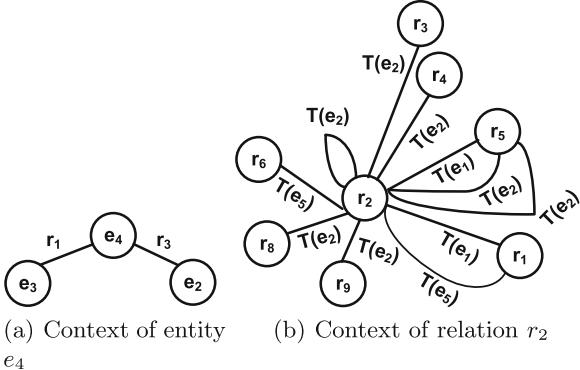


Fig. 3. Context of entity/relation. $T(e)$ denotes the type of entity e .

They are connected via entities of specific types, $T(e)$ denotes the type of entity e . For example, in *IMDB* the entity types could be actors, movies, directors, etc. Since each relation occurs many times in a KG, the same pair of relations can be connected via multiple entity types. Therefore, the context of a relation could be a multi-graph.

Since contexts of entities and relations are (multi)graphs having edge relations, the problem of context encoding is converted to relational subgraph encoding via R-GCN as follows.

$$h_i^{(l+1)} = \sigma \left(\sum_{r \in R} \sum_{j \in N_i^r} \frac{1}{|N_i^r|} W_r^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)} \right) \quad (1)$$

Here, $h_i^{(l)}$ is the hidden state of node v_i (from the context graph) at the l -th layer. N_i^r denotes the set of neighbor indices of node i under relation $r \in R$. Equation 1 aggregates transformed feature vectors of neighboring nodes and itself in a relation-specific manner and through a normalized sum, and then passes through an element-wise activation function $\sigma = \text{ReLU}$. We employ two R-GCNs, one for entities and another for relations. We consider one hidden layer in both R-GCNs since one hidden layer achieves better results and a good trade-off between accuracy and efficiency based on our experiments.

Joint Embedding. We aggregate the contextual subgraph embeddings of entities and relations with their knowledge embeddings via a gate strategy [35], and form the joint embedding \mathbf{o}^* of each object in the KG.

$$\mathbf{o}^* = \mathbf{g} \odot \mathbf{o}^k + (\mathbf{1} - \mathbf{g}) \odot \mathbf{sg}(o) \quad (2)$$

where \odot means element-wise multiplication, o is an entity or a relation, \mathbf{o}^k is its knowledge embedding, $\mathbf{sg}(o)$ is the vector representation of the context of o , $\mathbf{g} = \text{logistic}(\tilde{\mathbf{g}})$ ensures that the value of each element in the gate vector \mathbf{g} is in $[0, 1]$, and $\tilde{\mathbf{g}} \in \mathbb{R}^d$ is a parameter vector. Note that all entities share a \mathbf{g} denoted as \mathbf{g}^e , and all relations share another \mathbf{g} denoted as \mathbf{g}^r .

Embedding Learning. Given a triple (h, r, t) , we define a score function based on a translation operation as follows:

$$f(h, r, t) = \|\mathbf{h}^* + \mathbf{r}^* - \mathbf{t}^*\|_2^2 \quad (3)$$

where \mathbf{h}^* , \mathbf{r}^* and \mathbf{t}^* are computed by Eq. 2.

For training, we define a margin-based loss function:

$$\mathcal{L} = \sum_{(h, r, t) \in S} \sum_{(h', r, t') \in S'} \max(0, f(h, r, t) + \gamma - f(h', r, t')) \quad (4)$$

where γ is the margin, S is the set of correct triples, and S' is the set of incorrect triples. Since a KG only contains correct triples, we corrupt them by replacing head entities or tail entities to build S' . The replacement process follows the Bernoulli sampling method [34]. During training, the knowledge embeddings and context embeddings of all entities and relations are initialized following the uniform distribution $U(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ [5], where k is the number of dimensions for embeddings. All parameters including embeddings are updated using the Adam optimizer in each minibatch.

4 Online Embedding Updates

KGs are updated over time with addition and deletion of triples [9, 12, 19, 26]. KG embeddings should also be updated accordingly in an online manner. Following the inductive learning, we keep all the learnt parameters in R-GCNs and the gate strategy unchanged. Contextual element embeddings of existing entities and relations also remain the same. After a KG update, for many entities and relations, their contexts remain unaffected, so their contextual subgraph embeddings would remain uninterrupted. Thus, with existing knowledge embeddings of such entities and relations, corresponding triples would satisfy: $\mathbf{h}^* + \mathbf{r}^* \approx \mathbf{t}^*$. Hence, we also keep the knowledge embeddings of existing entities and relations unchanged so long as their contexts are unchanged.

For existing entities and relations with changed contexts, we do the following. Recall that the joint embedding of each entity or relation must approximately satisfy the translation relations in our model. This joint embedding depends on both the knowledge embedding and the contextual subgraph embedding. Consider an entity t that has changed context, so the vector representation of t 's context (i.e., $\mathbf{sg}(t)$, a combination of context element embeddings of its neighbor entities, computed by the R-GCN) would change to reflect this update. We next adjust the knowledge embedding of t (i.e., \mathbf{t}^k) to update its joint embedding \mathbf{t}^* , so that this updated \mathbf{t}^* approximately satisfies the translations in the modified graph. In practise, we find that the modifications happened in the joint embeddings are generally small due to local updates in a KG, which explains why our method is effective in approximately preserving the translations in the modified KG. In future, it would be interesting to analyze the errors due to approximately maintaining the translations in our online updates.

Finally, we also learn knowledge embeddings and contextual element embeddings of emerging entities and relations. In summary, we only need to compute

Table 1. Characteristics of datasets.

Datasets	#Entities (Avg.)	#Edges (Avg.)	#Relations (Avg.)	#Add triples (Avg.)	#Del triples (Avg.)	#Train (Avg.)	#Valid	#Test
YAGO-3SP	27 009	130 757	37	950	150	124 757	3 000	3 000
IMDB-3SP	169 146	524 296	14	9 181	521	518 296	3 000	3 000

the knowledge embeddings and contextual element embeddings of emerging entities and relations, as well as the knowledge embeddings of existing entities and relations with changed contexts, so that the joint embeddings of existing entities and relations with changed contexts also approximately satisfy the translations in the modified graph. With above strategies, the impacts of a KG update will be limited to a certain region, especially it will not affect the triples where the contexts of entities and relations are unchanged, which greatly improves efficiency. This also shows the benefit of having two separate vectors for every entity or relation in OUKE: knowledge embedding and context embedding.

Space Complexity. Let us denote by e , e_t , and r the total number of entities, entity types, and relations in the input KG. Assume each minibatch, on average, consists of e_b entities and r_b relations. We define the size of the adjacency matrix in the R-GCN for entities as $e_b \times e_b$ and that in the R-GCN for relations as $r_b \times r_b$ (as training happens in minibatches).

Suppose the dimension of the embedding space is k , the R-GCNs for entities and relations have l_e and l_r hidden layers, respectively, thus we have $\mathcal{O}((l_e \cdot r + l_r \cdot e_t)k \times k)$ weight matrices. In the gate strategy, all entities, relations, and weights also correspond to k -dimensional parameter vectors. In addition, each entity and each relation has two vector representations, so we totally have $(2e + 2r)k$ -dimensional vectors to represent entities and relations. In summary, the space complexity of OUKE is $\mathcal{O}(e_b^2 + r_b^2 + (l_e \cdot r + l_r \cdot e_t)k^2 + (e + r)k)$. As we discussed in Sect. 3, the number of hidden layers $l_e = l_r = 1$ in our implementation.

Time Complexity. For learning from scratch and online learning, we analyze their time complexities of updating parameters when given a triple pair $\{(h, r, t), (h', r, t')\}$ in a minibatch.

Since we adopt the negative sampling strategy proposed in [34], in the given triple pair, if $h \neq h'$, then $t = t'$; if $h = h'$, then $t \neq t'$. When given a triple pair in a minibatch, updating knowledge embeddings and context subgraph embeddings of three entities and a relation requires $\mathcal{O}(8k)$, where k is the dimension of the embedding space. Besides, updating the parameters in two R-GCNs and the gate strategy requires $\mathcal{O}((l_e \cdot r + l_r \cdot e_b)k^2)$ and $\mathcal{O}(2k)$, respectively. Since learning from scratch needs to update all of the above parameters, the total time complexity of updating parameters for given a triple pair $\{(h, r, t), (h', r, t')\}$ is $O(10k + (l_e \cdot r + l_r \cdot e_b)k^2)$.

In online learning, all parameters in two R-GCNs and the gate strategy are unchanged, and we only update the knowledge embeddings and context subgraph embeddings of emerging entities and relations, as well as the knowledge

embeddings of existing entities and relations with changed contexts. Thus, the total time complexity is $O(\mu k)$ ($1 \leq \mu \leq 8$), which reflects the efficiency of online learning. Besides, online learning also has much fewer triples to train when comparing with learning from scratch (as we demonstrated in Example 1), which is another reason for efficiency improvement.

5 Experimental Results

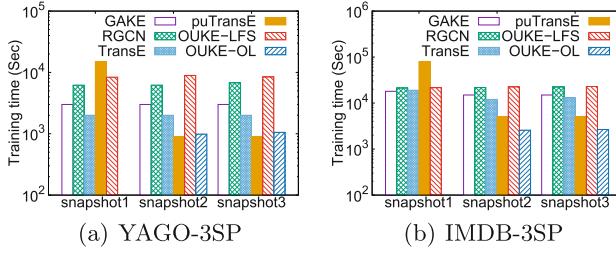
We conduct experiments to demonstrate the effectiveness and efficiency of OUKE in link prediction over evolving KGs. We employ PyTorch v1.1 deep learning library to implement OUKE and competitors. We perform experiments on a single machine with 256 GB, 2.2 GHz Intel(R) Xeon(R) CPU E5-2698 v4 processor. Our GPU platform is Tesla V100 (16 GB VRAM) with CUDA 9.0.

5.1 Experimental Setup

Datasets. We use two datasets from real-world KGs, having three snapshots. We split each snapshot into a training set, a validation set, and a test set. The three snapshots share the same validation set and test set, in which triples are unchanged in these snapshots.

Table 2. Accuracy results on link prediction. Lower values of MR and higher values of Hits@K (defined in Sect. 5.1) indicate better accuracy.

		YAGO-3SP		IMDB-3SP	
		MR	Hits@10	MR	Hits@10
Snapshot1	GAKE	2984	0.237	5798	0.213
	R-GCN	416	0.225	2646	0.112
	TransE	710	0.311	2408	0.352
	puTransE	938	0.262	3518	0.188
	OUKE-LFS	511	0.296	2987	0.349
Snapshot2	GAKE	3012	0.218	5542	0.218
	R-GCN	391	0.237	2482	0.123
	TransE	745	0.314	2314	0.340
	puTransE	897	0.259	3506	0.182
	OUKE-LFS	554	0.304	3018	0.364
	OUKE-OL	604	0.292	3133	0.329
Snapshot3	GAKE	2873	0.220	5623	0.219
	R-GCN	397	0.270	2871	0.052
	TransE	703	0.325	2575	0.351
	puTransE	1082	0.247	3522	0.187
	OUKE-LFS	537	0.305	3091	0.361
	OUKE-OL	599	0.281	3299	0.334

**Fig. 4.** Efficiency results on link prediction.

(1) YAGO-3SP. YAGO (<http://yago-knowledge.org/>) is a large-scale encyclopedic KG constructed from Wikipedia, WordNet, and GeoNames. Different versions of YAGO were published at various times. We extract subsets of YAGO2.5, YAGO3, and YAGO3.1 as three snapshots of our dataset YAGO-3SP.

(2) IMDB-3SP. The Internet Movie Database (IMDB) is a KG consisting of movies, TV series, actors, directors, etc., and their relations. IMDB provides daily dumps (<https://datasets.imdbws.com/>), and we download them from January 22 to January 24 in 2019 as three snapshots.

In Table 1, for each dataset we recorded: 1) the average numbers of entities (#Entities (Avg.)), edges (#Edges (Avg.)), and relations (#Relations (Avg.)) in different snapshots, respectively; 2) the average numbers of added triples (#Add Triples (Avg.)) and deleted triples (#Del Triples (Avg.)) between snapshots, respectively; 3) the average number of triples in the training sets (#Train (Avg.)) of different snapshots, the number of triples in the validation set (#Validate), and the number of triples in the test set (#Test).

Competitors. **(1) puTransE** [27] is the only existing model supporting online embedding learning for dynamic KGs. puTransE learns embeddings of entities and relations from local parts of a KG, so it avoids retraining on the entire KG, but cannot preserve the global structural information of the KG in the learnt embedding. **(2) TransE** [5] is a static KG embedding method using translation operations on entities and relations. **(3) GAKE** [10] simultaneously models triples and structural contexts in embedding learning on static KGs. **(4) R-GCN** [25] learns embedding considering multiple relations between entities in a static KG. Methods (2)-(4) learn embeddings only from scratch over static KGs.

Accuracy Metrics for Link Prediction. For each triple (h, r, t) in the test set, we replace the head entity h (or tail entity t) with each entity e in the snapshot to construct a triple (e, r, t) (or (h, r, e)), and rank all e based on the score calculated by the scoring function. If a constructed triple occurs in the training set, then the corresponding entity e will not participate in the ranking process, because training data cannot be used in testing. Based on such ranking results, we can get the rank of the original correct entity in each test triple, and we use the following evaluation metrics: **(1) Mean Rank (MR):** the average rank of all head entities and tail entities in test triples. **(2) Hits@K:** the proportion of

the ranks not larger than K for all head entities and tail entities in test triples. Lower values of MR and higher values of Hits@K indicate better accuracy.

5.2 Accuracy and Efficiency Results

Our empirical results in Table 2 indicate that our learning from scratch approach (OUKE-LFS) produces the best or second-best MR and Hits@10 scores over our datasets. Our online learning method (OUKE-OL) is quite competitive as well, however its accuracy is lower than OUKE-LFS because OUKE-OL retrains on a limited number of triples during online learning. On the other hand, OUKE-OL is faster. For instance, over IMDB-3SP, OUKE-OL is 2–3 orders of magnitude faster than static embedding methods, and up to an order of magnitude faster than puTransE (Fig. 4). Considering accuracy and efficiency trade-offs, we find OUKE-OL to be the most suitable for online updates of dynamic KG embedding.

6 Conclusions

We presented a context-aware dynamic knowledge graph (KG) embedding method OUKE, which not only learns embeddings from scratch, but also supports online embedding updates. Compared with state-of-the-art static and dynamic KG embedding techniques on dynamic datasets, OUKE has comparable effectiveness and much better efficiency in online learning. In future, it would be interesting to boost the accuracy of OUKE by improving its context embedding phase, analyzing its robustness with respect to repeated and batch updates, and deploying OUKE in more downstream applications such as KG-based question answering.

Acknowledgement. Arijit Khan is supported by MOE Tier1 and Tier2 grants RG117/19, MOE2019-T2-2-042, and a Delta Corporate Lab Grant SLE-RP8.

References

1. Source for IMDB dataset. <https://www.imdb.com/interfaces/>
2. Use Deep Search to Explore the COVID-19 Corpus. <https://www.research.ibm.com/covid19/deep-search/>
3. Ali, M., et al.: Bringing light into the dark: a large-scale evaluation of knowledge graph embedding models under a unified framework (2020). CoRR abs/2006.13365
4. Bollacker, K.D., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively Created Graph Database for Structuring Human Knowledge. In: SIGMOD (2008)
5. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NIPS (2013)
6. Chen, X., Chen, M., Fan, C., Uppunda, A., Sun, Y., Zaniolo, C.: Multilingual knowledge graph completion via ensemble knowledge transfer. In: EMNLP (Findings)

7. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: AAAI (2018)
8. Dasgupta, S.S., Ray, S.N., Talukdar, P.: HyTE: hyperplane-based temporally aware knowledge graph embedding. In: EMNLP (2018)
9. Dong, X.L.: Challenges and innovations in building a product knowledge graph. In: KDD (2018)
10. Feng, J., Huang, M., Yang, Y., Zhu, X.: GAKE: graph aware knowledge embedding. In: COLING (2016)
11. Gyrard, A., Gaur, M., Thirunarayanan, K., Sheth, A.P., Shekarpour, S.: Personalized health knowledge graph. In: CKGSemStats@ISWC (2018)
12. Hellmann, S., Stadler, C., Lehmann, J., Auer, S.: DBpedia live extraction. In: OTM Conferences (2009)
13. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: YAGO2: a spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell.* **194**(2013), 28–61 (2013)
14. Huang, X., Zhang, J., Li, D., Li, P.: Knowledge graph embedding based question answering. In: WSDM (2019)
15. Jin, J., Luo, J., Khemmarat, S., Gao, L.: Querying web-scale knowledge graphs through effective pruning of search space. *IEEE Trans. Parallel Distrib. Syst.* **28**(8), 2342–2356 (2017)
16. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
17. Khan, A., Wu, Y., Aggarwal, C.C., Yan, X.: NeMa: fast graph search with label similarity. *PVLDB* **6**(3), 181–192 (2013)
18. Lehmann, J., et al.: DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semant. Web* **6**(2), 167–195 (2015)
19. Lin, X., Li, H., Xin, H., Li, Z., Chen, L.: KBPearl: a knowledge base population system supported by joint entity and relation linking. *PVLDB* **13**(7), 1035–1049 (2020)
20. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI (2015)
21. Liao, S., Liang, S., Meng, Z., Zhang, Q.: Learning dynamic embeddings for temporal knowledge graphs. In: WSDM (2021)
22. Mitchell, T.M., et al.: Never-ending learning. *Commun. ACM* **61**(5), 103–115 (2018)
23. Nakashole, N., Tylenda, T., Weikum, G.: Fine-grained semantic typing of emerging entities. In: ACL (2013)
24. Nickel, M., Tresp, V., Kriegel, H.-P.: A three-way model for collective learning on multi-relational data. In: ICML (2011)
25. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: ESWC (2018)
26. Shin, J., Wu, S., Wang, F., Sa, C.D., Zhang, C., Ré, C.: Incremental knowledge base construction using DeepDive. *PVLDB* **8**(11), 1310–1321 (2015)
27. Tay, Y., Luu, A.T., Hui, S.C.: Non-parametric estimation of multiple embeddings for link prediction on dynamic knowledge graphs. In: AAAI (2017)
28. Trivedi, R., Dai, H., Wang, Y., Song, L.: Know-evolve: deep temporal reasoning for dynamic knowledge graphs. In: ICML (2017)
29. Trivedi, R., Farajtabar, M., Biswal, P., Zha, H.: DyRep: learning representations over dynamic graphs. In: ICLR (2019)
30. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: ICML (2016)

31. Wang, M., Qiu, L., Wang, X.: A survey on knowledge graph embeddings for link prediction. *Symmetry* **13**(3), 485 (2021)
32. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **29**(12), 2724–2743 (2017)
33. Wang, Y., Khan, A., Wu, T., Jin, J., Yan, H.: Semantic guided and response times bounded top-k similarity search over knowledge graphs. In: ICDE (2020)
34. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: AAAI (2014)
35. Xu, J., Qiu, X., Chen, K., Huang, X.: Knowledge graph representation with jointly structural and textual encoding. In: IJCAI (2017)
36. Yang, B., Yih, W.-T., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: ICLR (2015)
37. Zhao, Y., Zhang, A., Xie, R., Liu, K., Wang, X.: Connecting embeddings for knowledge graph entity typing. In: ACL (2020)
38. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.-Y.: Collaborative knowledge base embedding for recommender systems. In: KDD (2016)
39. Zhu, D., Cui, P., Zhang, Z., Pei, J., Zhu, W.: High-order proximity preserved embedding for dynamic networks. *IEEE Trans. Knowl. Data Eng.* **30**(11), 2134–2144 (2018)



High-Speed and Noise-Robust Embedding of Hypergraphs Based on Double-Centered Incidence Matrix

Shuta Ito^(✉) and Takayasu Fushimi

School of Computer Science, Tokyo University of Technology,
Hachioji 192-0982, Japan
{g21210088d,fushimity}@edu.teu.ac.jp

Abstract. In this study, for the purpose of robust clustering against noise, we propose a fast method of mapping hypernodes and hyperedges onto a unit hypersphere by focusing on the sparsity of the incidence matrix of a hypergraph. Our embedding method quantifies the relative strength of the relationship between a hypernode and a hyperedge from a double-centered incidence matrix, so that the weight of the noisy relationship can be suppressed to a small value, and the influence of noise on the embedding vectors of neighbor nodes and edges can be reduced. From the experimental evaluations using synthetic and real-world hypergraph data, we confirmed that our method outputs clustering results with good accuracy as well as a state-of-the-art method, especially for very noisy hypergraphs, and faster than other compared methods.

1 Introduction

In recent years, with the development of SNSs, the relationships that can be expressed in graphs are increasing around our lives. Since graphs are difficult to handle by numerical calculations including four arithmetic operations, one tries to solve various problems such as link prediction, node classification, and clustering relatively easily by expressing nodes and edges as vectors, and to do so is becoming mainstream. Many of these methods target generally-handled graphs that express the relationship between two nodes, and cannot be applied to hypergraphs straightforwardly. The hypergraph can represent the relationship of not only two nodes but also any number of nodes as shown in Fig. 1, and is attracting attention because it is more expressive than the graphs that are generally handled. In the following, graphs that are generally handled are referred to as normal graphs in order to distinguish them from hypergraphs.

Research on hypergraphs has been actively done in recent years [4, 16]. Among them, as in the case of normal graphs, there is a wide range of research on community extraction as known as graph clustering that extracts nodes groups where nodes are relatively densely connected, but there have not yet been established methods from the viewpoint of accuracy and computational complexity [7–10]. The clustering methods based on the maximization of hypergraph-modularity

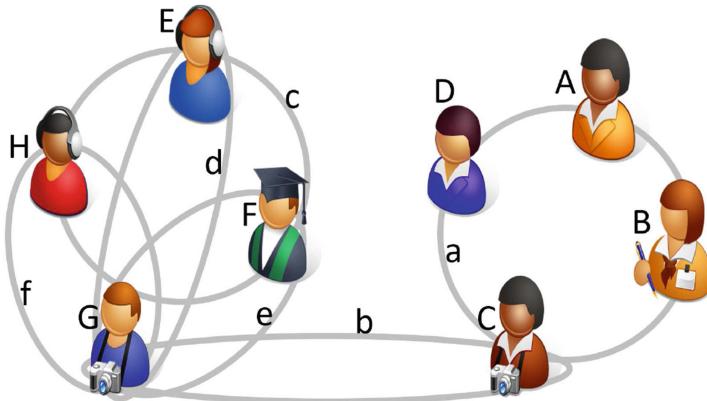


Fig. 1. Example of a hypergraph: a person with the uppercase alphabet is a hypernode, and a gray circle with the lowercase alphabet is a hyperedge.

[7–10] may not provide an appropriate solution depending on the structure of the hypergraph, especially for large-scale hypergraphs, a large amount of computation time is required as shown in [6].

Some embedding methods for normal graphs have been proposed [5, 11]. By applying these methods to the transformed graph according to the clique expansion that creates an edge between all two nodes contained in each hypergraph, it is possible to obtain a vector representation in principle. But the clique-expansion itself takes a lot of time when a large hyperedge is included, and the number of edges after conversion becomes enormous, so the sparsity that is advantageous in the calculation is somewhat impaired. In recent years, a method of directly embedding a hypergraph has also been proposed [12, 15, 16], but these methods also take a heavy computational load as shown in the experiments in this paper. In addition, the embedding method based on the eigenvectors of the graph Laplacian outputs a low-quality solution for hypergraphs with noise-edges, which exist in between nodes of different clusters. Therefore, the development of a high-speed and noise-robust embedding algorithm for hypergraphs is an important research topic.

In this study, for the purpose of robust clustering against noise, we propose a fast method of mapping hypernodes and hyperedges onto a hypersphere by focusing on the sparsity of the incidence matrix of a hypergraph. Our embedding method quantifies the relative strength of the relationship between a hypernode and a hyperedge from a double-centered incidence matrix, so that the weight of the noisy relationship can be suppressed to a small value, and the influence of noise on the embedding vectors of neighbor nodes and edges can be reduced. In our experimental evaluations using synthetic and real-world hypergraphs, the usefulness and efficiency of the proposed embedding method are evaluated from the viewpoint of the clustering problem.

2 Related Work

In this section, we explain existing research on graph embedding and graph clustering (community extraction), and clarify the position of this research.

2.1 Embedding

Based on the eigenvectors of the Laplacian matrix that represents the potential properties of the graph structure, a method called Spectral Embedding is famous as a method of producing a low-dimensional representation of nodes and is still widely used [11]. Furthermore, for arbitrary metric space objects, Laplacian Eigen Map constructs a neighborhood graph such as a k -Nearest Neighbor graph on the distance between objects and performs low-dimensional embedding based on its Laplacian matrix. For the graph consists of N nodes, the Laplacian matrix is a $N \times N$ matrix, and the cost of finding multiple eigenvectors by the iterative method is very high, so it may be difficult to calculate for a large-scale graph. Also, depending on the structure of the graph, the difference between the eigenvalues (eigen-gap) may be small and may not converge properly. In Spectral hypergraph embedding by Zhou et al. [16], the Laplacian matrix is calculated from the adjacency matrix of the normal graph obtained by clique expansion of the hypergraph, and then the eigenvectors are calculated to realize the embedding of the hypergraph. As shown in the later section of this paper, a low-quality solution is obtained for a graph containing noise.

In more recent years, following Word2vec, which has been successful in the field of natural language processing, Node2Vec has been proposed as a method for acquiring a vector representation of nodes [5]. This method regards a node-sequence obtained by a random walk starting from each node as sequence data which represents the relationship with neighboring nodes and learns the embedding vector based on Skip-gram used employed in Word2vec. Node2Vec is widely used because it has excellent expressive ability, but it is known that when the number of nodes and walks is large, a large number of iterations are required for convergence and the calculation cost is expensive.

In a method called NetVec proposed in [12], the hypergraph is converted into a bipartite graph, the vector of the hyperedge is defined by composing the vectors of the adjacent hypernode, and the vector of the hypernode is defined by composing the vectors of the adjacent hyperedges. Although NetVec is common with our method in that the vectors of hypernodes and hyperedges are fixed and updated alternately and these update processes are iteratively repeated, NetVec differs in that vectors are not normalized by their norm and that it considers the vector one iteration before by a certain rate.

As a method for learning the embedding representation of hypergraphs derived from Location Based Social Network, LBSN2Vec has been proposed [15]. In LBSN2Vec, as with Node2Vec, the relationship between hypernodes is regarded as sequence data based on a random walk on the graph, and an embedding representation is acquired. When learning the embedding vector, the regression line that maximizes the cosine similarity with each node vector is obtained

from the composite vector of the representation vectors of the nodes included in a certain hyperedge. Then, the embedding vector is learned so that the cosine similarity is the maximum for these nodes and the cosine similarity is the minimum for the negatively sampled nodes. It is common with our proposed method in that the embedding vector is obtained so that the cosine similarity between the vectors is maximized, but it differs from this study in that it is sequenced by a random walk and negative sampling is used. Our method attempts to embed both nodes and edges into the same Euclidean space as these methods [12, 15] do.

2.2 Clustering

Needless to say, many methods have been proposed for clustering for normal graphs, which cannot be listed here. The authors of work [14] pointed out that nodes with high degrees are more likely to connect to noise-edges which bridge multiple clusters. Based on this fact, a deep community detection method has been proposed, which identifies and removes noise-edges or noise-nodes which span many clusters, and repeatedly divides target graphs into subgraphs [1].

There some studies that generalize the modularity measure for normal graphs to apply to hypergraphs [7, 9]. Kumar et al. [9] calculated the modularity using the adjacency matrix in which the degree of each node was modified in response to degree increasing due to the clique expansion. To achieve maximum modularity, the method repeats two phase: the phase of fast clustering of nodes based on the Louvain method and the phase of updating the weight for each hyperedge. The weight of the hyperedge that spans some clusters in a well-balanced manner is reduced and updated so that it is easily cut in the next phase. The iteration is terminated when the update amount of the weight matrix becomes sufficiently small, and the clustering result is output.

Kaminski et al. [7] generalized the Chung-Lu model and defined an exact modularity measure based on the model. To achieve clustering results that maximizes the modularity, CNM [3] like algorithm is employed. However, due to a huge amount of computational costs, the authors done their experiments with an algorithm that randomly selects hyperedges and restores. On the other hand, the proposed method in this study is a high-speed method that takes advantage of the sparseness of the graph because it focuses on the incidence matrix of the hypergraph.

3 Proposed Method

In our proposed method, for a given hypergraph $H = (\mathcal{V}, \mathcal{E})$, hypernode $v \in \mathcal{V}$ and hyperedge $e \in \mathcal{E}$ are embedded into the same D -dimensional unit hypersphere and divide the hypernodes into K clusters. In this paper, the numbers of hypernodes and hyperedges are denoted as $N = |\mathcal{V}|$ and $M = |\mathcal{E}|$, respectively.

	a	b	c	d	e	f		a	b	c	d	e	f
A	1	0	0	0	0	0	A	0.65	-0.10	-0.23	-0.10	-0.10	-0.10
B	1	0	0	0	0	0	B	0.65	-0.10	-0.23	-0.10	-0.10	-0.10
C	1	1	0	0	0	0	C	0.48	0.73	-0.40	-0.27	-0.27	-0.27
D	1	0	0	0	0	0	D	0.65	-0.10	-0.23	-0.10	-0.10	-0.10
E	0	0	1	1	0	0	E	-0.52	-0.27	0.60	0.73	-0.27	-0.27
F	0	0	1	0	1	0	F	-0.52	-0.27	0.60	-0.27	0.73	-0.27
G	0	1	0	1	1	1	G	-0.85	0.40	-0.73	0.40	0.40	0.40
H	0	0	1	0	0	1	H	-0.52	-0.27	0.60	-0.27	-0.27	0.73

(a) Incidence matrix.

(b) Double-centered incidence matrix.

Fig. 2. Incidence matrix of example hypergraph.

3.1 Embedding

Our proposed method attempts to embed hypernodes \mathcal{V} and hyperedges \mathcal{E} into the same D -dimensional unit hypersphere such that adjacent hypernodes and hyperedges are to point the same direction and otherwise opposite one. We denote the embedding vector of hypernode v and hyperedge e as \mathbf{x}_v and \mathbf{y}_e , respectively. That is, our method emplaces the vectors \mathbf{x}_v and \mathbf{y}_e with a small angle on a hypersphere if v and e connected with strong relationship.

The connection relationship between hypernodes and hyperedges is represented by a $N \times M$ incidence matrix $\mathbf{B} = [b_{v,e}]_{v \in \mathcal{V}, e \in \mathcal{E}}$ whose element $b_{v,e}$ is 1 if v is included by e , and otherwise 0. In order to smooth out the noise such as nodes and edges with extremely high degree, we calculate a double-centered incidence matrix $\tilde{\mathbf{B}} = \mathbf{J}_N \mathbf{B} \mathbf{J}_M = [\tilde{b}_{v,e}]_{v \in \mathcal{V}, e \in \mathcal{E}}$ in the same way as the Young Householder transformation in the multidimensional scaling [13], by defining two centering matrices $\mathbf{J}_N = \mathbf{I}_N - \frac{1}{N}\mathbf{E}_N$ and $\mathbf{J}_M = \mathbf{I}_M - \frac{1}{M}\mathbf{E}_M$ where \mathbf{I}_N and \mathbf{I}_M stand for $N \times N$ and $M \times M$ identity matrices, respectively, and \mathbf{E}_N and \mathbf{E}_M are $N \times N$ and $M \times M$ matrices whose elements are all one. The value of $\tilde{b}_{v,e}$ get to be positive if the hypernode v and hyperedge e have an connectivity, otherwise negative. The closer the value is to +1, the stronger the connection, and the closer it is to -1, the weaker the relationship. Furthermore, hypernodes with a high degree and hyperedges with a large size tend to have a noisy relationship, but by making their weights close to 0, they are less likely to be considered when calculating the embedded vector. In case of a hypergraph shown in Fig. 1, the incidence matrix and double-centered incidence matrix get to be like Fig. 2.

Hypernodes and hyperedges with strong ties should be embedded in the same direction from the origin on the hypersphere, *i.e.*, the cosine similarity, which is the inner product between embedding vectors with a norm of 1, should get

close to 1. On the contrary, the embedding vectors of hypernodes and hyperedges, which are weakly related, are embedded in the opposite direction from the origin, *i.e.*, the cosine similarity should get close to -1 . Therefore, the embedding vector is calculated so that the cosine similarity between the embedded vectors matches the sign of the elements of the double-centered connection matrix.

Given embedding vectors $\mathbf{X} = [\mathbf{x}_v]_{v \in \mathcal{V}}$, the embedding vector of e that has the highest cosine similarity to the node vectors with consideration of weights $\tilde{b}_{v,e}$ is calculated as follows:

$$\mathbf{y}_e \leftarrow \arg \max_{\mathbf{y} \in [-1,1]^D, \|\mathbf{y}\|=1} \sum_{v \in \mathcal{V}} \tilde{b}_{v,e} \cos(\mathbf{y}, \mathbf{x}_v) = \sum_{v \in \mathcal{V}} \tilde{b}_{v,e} \mathbf{x}_v.$$

Similarly, given embedding vectors $\mathbf{Y} = [\mathbf{y}_e]_{e \in \mathcal{E}}$, the embedding vector of v that has the highest cosine similarity to the edge vectors with consideration of weights $\tilde{b}_{v,e}$ is calculated as follows:

$$\mathbf{x}_v \leftarrow \arg \max_{\mathbf{x} \in [-1,1]^D, \|\mathbf{x}\|=1} \sum_{e \in \mathcal{E}} \tilde{b}_{v,e} \cos(\mathbf{x}, \mathbf{y}_e) = \sum_{e \in \mathcal{E}} \tilde{b}_{v,e} \mathbf{y}_e.$$

Therefore, the embedding vectors of the hypernodes and the hyperedges can be obtained according to the following update formula:

$$\mathbf{y}_e \leftarrow \frac{\tilde{\mathbf{y}}_e}{\|\tilde{\mathbf{y}}_e\|}, \quad \tilde{\mathbf{y}}_e \leftarrow \sum_{v \in \mathcal{V}} \tilde{b}_{v,e} \mathbf{x}_v, \quad (1)$$

$$\mathbf{x}_v \leftarrow \frac{\tilde{\mathbf{x}}_v}{\|\tilde{\mathbf{x}}_v\|}, \quad \tilde{\mathbf{x}}_v \leftarrow \sum_{e \in \mathcal{E}} \tilde{b}_{v,e} \mathbf{y}_e. \quad (2)$$

In this update formula, the embedding vectors of the hypernodes in a hyperedge are computed by the embedding vector of that hyperedge, so the adjacency relationship between hypernodes is indirectly considered via the hyperedge.

Now, we define the set of hyperedges that include hypernode v as $\Gamma(v) = \{e \in \mathcal{E} | v \in e\}$. In our algorithm shown below, after initializing $N + M$ D -dimensional vectors, the processing of centering, vector composition, centering, and normalization is repeated until it converges:

1. Initializing: $\mathbf{x}_v \leftarrow [-1,1]^D$, $\mathbf{y}_e \leftarrow [-1,1]^D$;
2. Normalizing: $\mathbf{x}_v \leftarrow \frac{\mathbf{x}_v}{\|\mathbf{x}_v\|}$, $\mathbf{y}_e \leftarrow \frac{\mathbf{y}_e}{\|\mathbf{y}_e\|}$;
3. Repeating until convergence:
 - (a) Centering: $\tilde{\mathbf{x}}_v \leftarrow \mathbf{x}_v - \frac{1}{N} \sum_{v' \in \mathcal{V}} \mathbf{x}_{v'}$;
 - (b) Composing: $\mathbf{y}_e \leftarrow \sum_{v \in e} \tilde{\mathbf{x}}_v$;
 - (c) Centering: $\tilde{\mathbf{y}}_e \leftarrow \mathbf{y}_e - \frac{1}{M} \sum_{e' \in \mathcal{E}} \mathbf{y}_{e'}$;
 - (d) Normalizing: $\mathbf{y}_e \leftarrow \frac{\tilde{\mathbf{y}}_e}{\|\tilde{\mathbf{y}}_e\|}$;
 - (e) Centering: $\tilde{\mathbf{y}}_e \leftarrow \mathbf{y}_e - \frac{1}{M} \sum_{e' \in \mathcal{E}} \mathbf{y}_{e'}$;
 - (f) Composing: $\mathbf{x}_v \leftarrow \sum_{e \in \Gamma(v)} \tilde{\mathbf{y}}_e$;
 - (g) Centering: $\tilde{\mathbf{x}}_v \leftarrow \mathbf{x}_v - \frac{1}{N} \sum_{v' \in \mathcal{V}} \mathbf{x}_{v'}$;
 - (h) Normalizing: $\mathbf{x}_v \leftarrow \frac{\tilde{\mathbf{x}}_v}{\|\tilde{\mathbf{x}}_v\|}$;

At the step 3-(a), 3-(b), 3-(c) and 3-(d) corresponds to the update formula (1), and 3-(e), 3-(f), 3-(g), and 3-(h) corresponds to the update formula (2). In our algorithm, in order to avoid the $N \times M$ summations of D -dimensional vectors according to the update formula (1) and (2), a centering operation is done to the embedding vectors instead of calculating the double-centered incidence matrix $\tilde{\mathbf{B}}$, for keeping the sparsity. Since each hypernode and hyperedge is represented by a D -dimensional vector, the spatial complexity is $O(D(N + M))$.

3.2 Clustering

Our proposed method aims to embed hypergraphs onto a hypersphere so as to obtain good clustering, thus we employ the spherical k -means method as a good fit clustering means.

Given the embedding vectors of hypernodes \mathbf{X} , the spherical k -means divides them into K clusters according to the pairwise similarity defined by $\rho(u, v) = \cos(\mathbf{x}_u, \mathbf{x}_v)$. Let denote K centroid vectors as $\mathbf{R} = [\mathbf{r}_1, \dots, \mathbf{r}_K]$ and a set of nodes whose nearest neighbor is a centroid \mathbf{r}_k as $\mathcal{V}_k = \{v; \mathbf{r}_k = \arg \max_{\mathbf{r} \in \mathbf{R}} \cos(\mathbf{r}, \mathbf{x}_v)\}$, the method determines the cluster $\mathcal{C} = \{\mathcal{V}_1, \dots, \mathcal{V}_K\}$ by the iterative improvement method so that the following objective function is maximized:

$$F(\mathcal{C}; \mathbf{X}) = \sum_{k=1}^K \sum_{v \in \mathcal{V}_k} \cos(\mathbf{r}_k, \mathbf{x}_v).$$

4 Experimental Settings

In this study, we evaluate our embedding method in terms of clustering, compared to conventional embedding and clustering methods.

4.1 Datasets

In our experiments, we utilized artificially created hypergraphs and real-world hypergraphs. As for artificial hypergraphs, we generated according to the method in the experiment of [2]. First, we give the number of hypernodes N , the number of hyperedges M , the number of clusters K , and the minimum/maximum number of hypernodes, s_{min} and s_{max} , contained in each hyperedge. Next, we assign a cluster number to each hypernode. Then, we create M hyperedges as follows:

1. Decide the size s of hyperedge e according to the Weibull distribution¹ $s \sim W(s; \lambda, \eta) = \frac{\eta}{\lambda} \left(\frac{s}{\lambda}\right)^{\eta-1} \exp\left(-\left(\frac{s}{\lambda}\right)^\eta\right)$;
2. Decide whether hyperedge e of size s would be closed in a cluster or lie across multiple clusters, according to the reverse sigmoid distribution $Q(s; \alpha, \delta) \propto \frac{1}{1 + \exp(\alpha s - \delta)}$;
3. Decide hypernodes which hyperedge e contain;

¹ In [2], the hyperedge size follows the uniform distribution.

At the step 3, as for the hyperedge that is closed in one cluster, a cluster is randomly selected from K clusters and s hypernodes are randomly selected from the selected cluster. As for the hyperedge that lies across multiple clusters, s hypernodes are selected from all the ones.

In our experiments, we set the parameters as $N = 1000, M = 500, K = 10, s_{min} = 2, s_{max} = 10, \lambda = 5, \eta = 1, \delta = 2$. As for the noise-level parameter α in the reverse sigmoid distribution, we change the value in the range of $\{0.1, 0.2, \dots, 1.0\}$ and generate 10 hypergraphs. The larger value of α is, the more a hyperedge would lie across multiple clusters, so that the generated hypergraph would have a weak community structure and the clustering get to be harder. Hereafter, the generated artificial hypergraphs are referred to as Chodrow (α).

As for a real-world hypergraph, we employed DBLP bibliographic dataset², and regarded the authors as hypernodes and the set of authors who wrote a single paper together as hyperedges. We extracted and utilized its largest connected component for our experiments. The numbers of hypernodes and hyperedges are 3,965 and 5,596, respectively.

4.2 Evaluation Measure

We employed the following two measures to evaluate the clustering accuracy.

- Adjusted Rand Index

ARI is an index to measure how similar the two clustering results are. Let $c^*(u)$ and $\hat{c}(u)$ be the actual cluster id and the estimated cluster id for node u , respectively. By defining the following sets of node pairs:

$$\begin{aligned}\mathcal{S}_1 &= \{(u, v) \in \mathcal{V} \times \mathcal{V}; c^*(u) = c^*(v)\}, \quad \mathcal{S}_2 = \{(u, v) \in \mathcal{V} \times \mathcal{V}; \hat{c}(u) = \hat{c}(v)\}, \\ \mathcal{D}_1 &= \{(u, v) \in \mathcal{V} \times \mathcal{V}; c^*(u) \neq c^*(v)\}, \quad \mathcal{D}_2 = \{(u, v) \in \mathcal{V} \times \mathcal{V}; \hat{c}(u) \neq \hat{c}(v)\}, \\ \mathcal{S} &= \{(u, v) \in \mathcal{V} \times \mathcal{V}; c^*(u) = c^*(v) \wedge \hat{c}(u) = \hat{c}(v)\}, \\ \mathcal{D} &= \{(u, v) \in \mathcal{V} \times \mathcal{V}; c^*(u) \neq c^*(v) \wedge \hat{c}(u) \neq \hat{c}(v)\}, \\ \mathcal{L} &= \{(u, v) \in \mathcal{V} \times \mathcal{V}\},\end{aligned}$$

and the following probabilities: $p_1 = |\mathcal{S}_1|/|\mathcal{L}|$, $p_2 = |\mathcal{S}_2|/|\mathcal{L}|$, $q_1 = 1 - p_1 = |\mathcal{D}_1|/|\mathcal{L}|$, $q_2 = 1 - p_2 = |\mathcal{D}_2|/|\mathcal{L}|$, ARI is calculated as follows:

$$ARI = \frac{|\mathcal{S}| + |\mathcal{D}| - |\mathcal{L}|(p_1 \cdot p_2 + q_1 \cdot q_2)}{|\mathcal{L}| - |\mathcal{L}|(p_1 \cdot p_2 + q_1 \cdot q_2)}.$$

A high value of ARI means that the clustering result is close to the actual cluster.

- Hypergraph modularity [8]

Hypergraph modularity is proposed by Kaminski et al., which is an index that measures the goodness of graph division, and is calculated by the ratio

² <https://dblp.uni-trier.de/xml/>.

of the number of edges closed in a cluster:

$$Q(\mathcal{V}_1 \dots \mathcal{V}_k) = \sum_{s \geq 2} \sum_{c=\lfloor s/2 \rfloor + 1}^s w_{c,s} q_{c,s}(\mathcal{V}_1 \dots \mathcal{V}_k). \quad (3)$$

Here, $q_{c,s}$ is the partial modularity, which consider only the case where just on c hypernodes of a hyperedge of size s are included in one cluster. By denoting the sum of the degree of hypernodes belonging to \mathcal{V}_k as $\text{vol}(\mathcal{V}_k)$ and the probability of getting exactly c successes in s independent Bernoulli trials under the success probability p as $\Pr[\text{Bin}(s, p) = c]$, the $q_{c,s}$ is defined as follows:

$$q_{c,s}(\mathcal{V}_1 \dots \mathcal{V}_k) = \frac{1}{M} \sum_{k=1}^K \left\{ M_{s,c}(\mathcal{V}_k) - M_s \cdot \Pr \left[\text{Bin} \left(s, \frac{\text{vol}(\mathcal{V}_k)}{\text{vol}(\mathcal{V})} \right) = c \right] \right\},$$

where the first term in a brace, $M_{s,c}(\mathcal{V}_k)$, is the actual number of hyperedges of size s that have exactly c hypernodes in a cluster \mathcal{V}_k , and the second one is the expected number of hyperedges that do the same in the corresponding random graph. In Eq. (3), $w_{c,s} \in [0, 1]$ is a weight for the partial modularity $q_{c,s}$. In this study, we adopt the following indices:

- Strict modularity: Considering only the case where all the hypernodes of a hyperedge are included in one cluster, *i.e.*, $w_{c,s} = 1$ if $c = s$, otherwise $w_{c,s} = 0$;
- Majority modularity: Equally considering the case where a majority of hypernodes is included in one cluster, *i.e.*, $w_{c,s} = 1$ if $c \geq \lfloor s/2 \rfloor + 1$;
- Generalized modularity: More considering the case where more hypernodes of a hyperedge are included in one cluster, *i.e.*, $w_{c,s} = c/s$;

Hypergraph Majority Modularity (HMM) is used as the evaluation index.

4.3 Methods Used for Comparison

As compared method, we employed graph embedding methods and hypergraph clustering methods shown below.

– KPPST [7,8]

Kaminski et al. generalized a notion of modularity for normal graphs to for hypergraphs, and proposed a clustering method based on the maximization of the hypergraph modularity. In [7], a maximization algorithm like CNM method [3] was presented that randomly selects hyperedges and repeatedly merges the two clusters which realize the maximum Q . In this study, we employed three methods each of which attempts to maximize the strict, the majority, and the generalized hypergraph modularity proposed in [8].

– Spectral clustering [11]

Spectral clustering is one of the representative clustering methods for normal graphs, which regards eigenvectors of the Laplacian matrix as a low-dimensional representation of each node, and clustering them by the standard

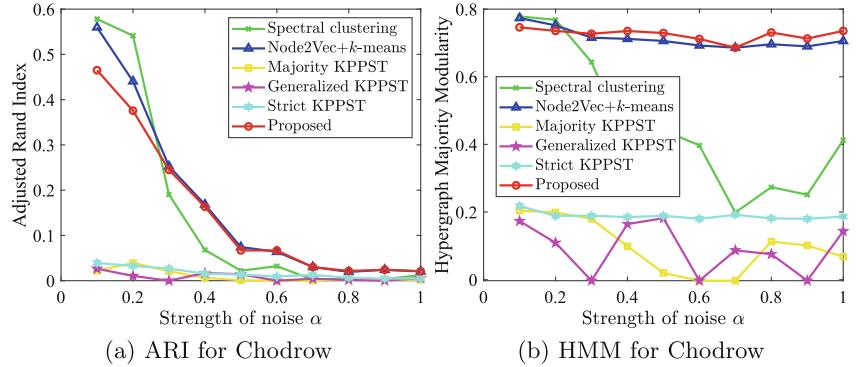


Fig. 3. w.r.t. the degree of strength of noise α .

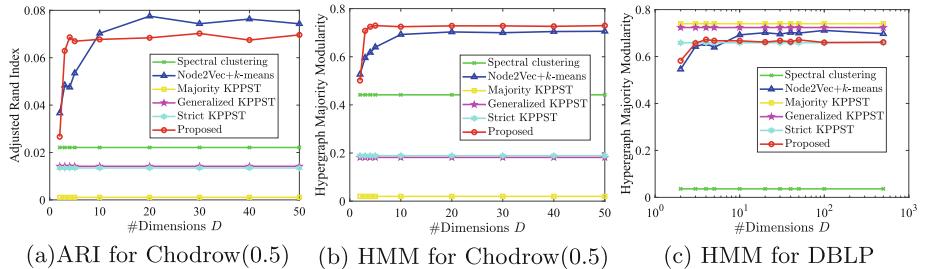


Fig. 4. w.r.t embedding dimension D .

k -means method. In this study, we constructed the symmetric normalized Laplacian matrix using the degree-preserving weighted graph [10], which is calculated from the incidence matrix of the hypergraph.

– Node2Vec [5]

Node2Vec is an algorithm that regards the nodes of the normal graph as words and the sequence of neighboring nodes visited by random walk starting from each node as sentences in the document, and embeds them using Word2Vec. In this study, for each node, 10 sequences were generated whose length was set to 30.

5 Evaluation Results

5.1 Quality of Clustering

Figure 3 shows the values of Adjusted Rand Index (ARI) and Hypergraph Modularity (HMM), when changing the parameter α , which stands for the degree of strength of noise, that is, the degree of weakness of community structure in artificial hypergraphs. For our proposed method and Node2Vec, the algorithm

was conducted 10 times and plotted the average values. In Fig. 3, the most accurate evaluation value among the clustering results obtained by embedding in a vector of 3 to 50 dimensions is plotted. From these results, at both the ARI and HMM, it can be seen that although Spectral clustering and Node2Vec can produce clustering results with better accuracy than the proposed method in the hypergraph with weak noise (around $\alpha < 0.3$), our proposed method clusters more accurately even in a hypergraph with strong noise (around $\alpha \geq 0.3$). In particular, Spectral clustering is notably less accurate in hypergraphs with strong noise.

Figure 4 depicts the values of ARI and HMM, when changing the number of dimensions D in embedding methods. The clustering methods, which are not based on embedding, are plotted as a constant. From Fig. 4, we can observe that our proposed method can accurately represent the hypergraph even with a low-dimensional vector compared to Node2Vec. Since the computational complexity of the proposed algorithm is proportional to the number of embedded dimensions D , it can be said to be a high-speed algorithm. Furthermore, we can say that the proposed method output clustering results for actual data with the same accuracy as existing hypergraph clustering methods.

5.2 Efficiency of Embedding

Table 1 indicates the execution time of the three embedding methods. In Spectral clustering (Spec), according to the convention, the embedding dimension D is set to the number of clusters K . In Node2Vec and the proposed method, according to the highest accuracy reported in Subsect. 5.1, D is set to 50 and 5, respectively.

In Table 1, there is no remarkable difference in execution time for artificial data with a small graph size, but there is a significant difference in execution time for large real data. From this, it can be said that the proposed method has the same level of accuracy as a typical existing embedding method, and is highly efficient in outputting results at high speed even for large-scale data.

Table 1. Execution time (sec.)

Dataset	Spec	Node2Vec	Proposed
#dimensions D	K	50	5
Chodrow ($\alpha = 0.1$)	5.49	6.23	6.21
Chodrow ($\alpha = 0.5$)	5.33	5.74	6.87
DBLP	173.51	205.91	22.52

6 Conclusion

In this study, we proposed an embedding method for hypergraphs with the aim of accurately clustering noisy hypergraphs that contain many hyperedges that

span between clusters. By double-centering the incidence matrix of the hypergraph, the strength of the relative relationship between the hypernode and the hyperedge is quantified, and the influence on the embedding vector of the noisy node or edge is kept small. In addition, we proposed an efficient algorithm that can be calculated without impairing the sparsity of the incidence matrix by centering the embedding vector instead of directly centering the incidence matrix. In the evaluation experiment using artificial data and real data, it was confirmed that the embedding vector and clustering result can be output faster with the same accuracy as Node2Vec, which is still widely used.

Future tasks include comparison with more graph and hypergraph embedding algorithms and evaluation of embedding accuracy from a viewpoint other than clustering.

Acknowledgments. This material is based upon work supported by JSPS Grant-in-Aid for Scientific Research (C) (No. 20K11940) and Early-Career Scientists (No. 19K20417).

References

1. Chen, P.Y., Hero, A.O.: Deep community detection. *IEEE Trans. Sig. Process.* **63**(21), 5706–5719 (2015)
2. Chodrow, P.S., Veldt, N., Benson, A.R.: Generative hypergraph clustering: from blockmodels to modularity. *Sci. Adv.* **7**(28), 1–14 (2021). <https://advances.sciencemag.org/content/7/28/eabh1303>
3. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. *Phys. Rev. E* **70**(6), 1–6 (2004). <https://doi.org/10.1103/PhysRevE.70.066111>
4. Do, M., Yoon, S., Hooi, B., Shin, K.: Structural patterns and generative models of real-world hypergraphs. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2000, pp. 176–186. Association for Computing Machinery, New York (2000). <https://doi.org/10.1145/3394486.3403060>
5. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016, pp. 855–864. Association for Computing Machinery, New York (2016). <https://doi.org/10.1145/2939672.2939754>
6. Ito, S., Fushimi, T.: Fast clustering of hypergraphs based on bipartite-edge restoration and node reachability. In: Proceedings of the 22nd International Conference on Information Integration and Web-Based Applications & Services, iiWAS 2020, pp. 115–124, Association for Computing Machinery, New York (2020). <https://doi.org/10.1145/3428757.3429095>
7. Kamiński, B., Poulin, V., Pralat, P., Szufel, P., Théberge, F.: Clustering via hypergraph modularity. *PLOS ONE* **14**(11), 1–15 (2019). <https://doi.org/10.1371/journal.pone.0224307>
8. Kamiński, B., Prałat, P., Théberge, F.: Community detection algorithm using hypergraph modularity. In: Benito, R.M., Cherifi, C., Cherifi, H., Moro, E., Rocha, L.M., Sales-Pardo, M. (eds.) *Complex Networks & Their Applications IX. COMPLEX NETWORKS 2020. Studies in Computational Intelligence*, vol. 943. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-65347-7_13

9. Kumar, T., Vaidyanathan, S., Ananthapadmanabhan, H., Parthasarathy, S., Ravindran, B.: Hypergraph clustering: a modularity maximization approach. CoRR abs/1812.10869 (2018). <http://arxiv.org/abs/1812.10869>
10. Kumar, T., Vaidyanathan, S., Ananthapadmanabhan, H., Parthasarathy, S., Ravindran, B.: Hypergraph clustering by iteratively reweighted modularity maximization. *Appl. Netw. Sci.* **5**(1), 1–22 (2020). <https://doi.org/10.1007/s41109-020-00300-3>
11. von Luxburg, U.: A tutorial on spectral clustering. *Stat. Comput.* **17**(4), 395–416 (2007)
12. Maleki, S., Wall, D., Pingali, K.: NetVec: a scalable hypergraph embedding system (March 2021). <https://arxiv.org/abs/2103.09660>
13. Torgerson, W.: Multidimensional scaling: I. Theory and method. *Psychometrika* **17**, 401–419 (1952). <https://doi.org/10.1007/BF02288916>
14. Wen, H., Leicht, E.A., D’Souza, R.M.: Improving community detection in networks by targeted node removal. *Phys. Rev. E* **83**, 016114 (2011). <https://doi.org/10.1103/PhysRevE.83.016114>
15. Yang, D., Qu, B., Yang, J., Cudre-Mauroux, P.: Revisiting user mobility and social relationships in LBSNs: a hypergraph embedding approach. In: The World Wide Web Conference, WWW 2019, pp. 2147–2157. Association for Computing Machinery, New York (2019). <https://doi.org/10.1145/3308558.3313635>
16. Zhou, D., Huang, J., Schölkopf, B.: Learning with hypergraphs: clustering, classification, and embedding. In: Proceedings of the 19th International Conference on Neural Information Processing Systems, NIPS 2006, pp. 1601–1608. MIT Press, Cambridge (2006)



Simple Negative Sampling for Link Prediction in Knowledge Graphs

Md Kamrul Islam^(✉), Sabeur Aridhi, and Malika Smail-Tabbone

Universite de Lorraine, CNRS, Inria, LORIA, 54000 Nancy, France
{kamrul.islam,sabeur.aridhi,malika.smail}@loria.fr

Abstract. Knowledge graph (KG) embedding methods learn the low dimensional vector representations of entities and relations of a knowledge graph, facilitating the link prediction task in knowledge graphs. During learning of embeddings, sampling negative triples is important because KGs have only observed positive triples. To the best of our knowledge, uniform-random, generative adversarial network (GAN)-based, and NSCaching, structure aware negative sampling (SANS) are four negative sampling methods in the literature. Unfortunately, they suffer from computational and memory inefficiency problems. In addition, their prediction performance are affected by the ‘vanishing gradient’ problem because of poor quality of sampled negative triples. In this paper, we propose a simple negative sampling (SNS) method based on the assumption that the entities which are closer in the embedding space to the corrupted entity are able to provide high-quality negative triples. Furthermore SNS has a good exploitation potential as it uses sampled high-quality negatives for improving the quality of negative triples in next steps. We evaluate our sampling method through link prediction task on five well-known knowledge graph datasets, WN18, WN18RR, FB15K, FB15K-237, YAGO3-10. The method is also evaluated on a new biological KG dataset (FIGHT-HF-23R). Experimental results show that the SNS improves the prediction performance of KG embedding models, and outperforms the existing sampling methods.

Keywords: Knowledge graph embedding · Link prediction · Negative sampling

1 Introduction

A knowledge graph (KG) is a graph-based representation of knowledge which illustrates real-world entities and their relations covering various domains [1]. Formally, a KG is represented as a collection of RDF triples, (head, relation, tail) where the head and the tail are two entities which are connected by a specific relation, e.g. (Shakespeare, isAuthorOf, Hamlet). KGs are fundamental building blocks for many applications, ranging from question answering to content-based recommendation. Some of the notable KGs are FreeBase, WikiData, DBPedia, Yago, NELL. Generally, KGs contain millions of entities and billions of triples.

Despite of their huge size, the incompleteness of KGs is well known. For example, birth place of more than 70% of person entities in Freebase is missing [2]. The incompleteness issue of KGs motivates researchers to study on how to add new triples in KGs. This task is known as link prediction which infer new triples based on the observed triples in KGs.

In recent years, many researchers have developed embedding models to learn vector representations (or embeddings) entities and relations in KGs. These models perform the link prediction task based on the learned embeddings. The training of these models requires positive triples as well as negative/non-observed triples. However, only positive triples are available in KGs. Importantly, the quality of negative triples does matter [3, 4]. This statement brings the importance of sampling negative triples. Unfortunately, this important perspective of embedding model is less focused in the literature. One way to sample negatives is designed based on ‘closed-world’ assumption where all of the non-observed triples are necessarily false and used as negatives. However, this assumption is not entirely true for KGs due to their incompleteness [5]. Alternatively, most of the KG embedding models samples negatives from non-observed triples under ‘open-world’ assumption where a non-observed triple may be positive or negative. To the best of our knowledge, there exist four negative sampling methods: uniform-random [6], GAN-based [3, 7, 8], NSCaching [4], and SANS [9]. However, each of them has its own pros-cons and the current state-of-art still lacks a good negative sampling method. In this article, we propose a simple but efficient method called SNS to sample high-quality negative triples in KG. In sampling, the trade-off between exploration and exploitation is crucial in searching for a high-quality samples [10]. Exploration corresponds to the capacity of the sampling method to select high-quality negative triples from unexplored areas whereas exploitation favours the utilization of already known negative triples to sample other negatives. SNS makes a good balance between exploration and exploitation to improve the quality of negative triples. We designed SNS as general sampling method that can be plugged to any KG embedding model for link prediction.

2 State-of-Art

High-quality negative triples, which are not readily available, contribute during training of a KG embedding model. Generally, candidate negative triple set is generated by positive triple perturbation where head/tail entities are replaced with other entities [5] and then a sampler samples negative triples from the set. ‘Uniform-random’ is the mostly used negative sampling method where negative triples are randomly sampled from a uniform distribution of candidate negatives [4]. Though ‘uniform-random’ is simple, a sampled negative triple could be completely unrelated to the corresponding positive triple and is easily classified as negative. Consequently, ‘uniform-random’ method seriously suffers from ‘zero-loss’/‘vanishing-gradient’ problem [3]. Recently, the generative property of GAN frameworks to generate high-quality negatives for avoiding the

‘vanishing-gradient’ problem is studied. The generator of GAN is trained to pick high-quality negative triples whereas the discriminator part is trained to learn embeddings. IGAN [8], KBGAN [3], KSGAN [7] are three existing GAN-based sampling methods for KGs. Compared to ‘uniform-random’ sampling, these methods improve the quality of negative triples undoubtedly. However, these methods increase the number of model parameters and take extra costs on training for parameter optimization [4]. In addition, they suffer from instability and degeneracy problems because of adopting the complex reinforcement learning to train the generator [4]. To avoid the excessive training time of GAN-based methods, Zhang et al. [4] proposed a ‘distilled’ version of GAN-based methods, namely NSCaching which stores negatives with high scores in head and tail caches for each positive triple, and then samples negatives directly from the caches. With NSCaching sampling, KG embedding models show competitive link prediction performance. However, the memory requirement increases exponentially with the size of KGs. Also, the regular updating of caches increases the computational time. Thus, scalability is a big issue for NSCaching and it is not recommended for large KGs. Apart from the above-mentioned sampling methods, the hard negative mining in contrastive learning motivates Ahrabian et al. [9] to study the neighborhood information of entities to sample negatives. They develop SANS method based on an assumption that neighbor entities without direct relation are good candidates for generating negatives. However, the one-time generation of negatives prior to the start of the embedding learning process is expensive in terms of memory requirements, as it requires the whole adjacency matrix of a KG in main memory.

3 The Proposed SNS Method

For link prediction, KG embedding models are trained with positive and negative triples to learn embeddings of entities and relations. In this section, we first briefly describe a classical KG embedding model, and then our proposed SNS negative sampling method. Throughout the paper, we use \mathbb{E} to denote the set of all entities, \mathbb{R} to denote the set of all relations, \mathbb{S} to denote the set of positive training triples, \mathbb{D} to denote the set of positive test triples, \mathbb{Q} to denote the set of all positive triples, d to denote embedding dimension size, m to denote the batch size, S_m and S'_m to denote a batch of positive and respective negative triples. The architecture of a classical KG embedding model is given in Fig. 1 which starts with initializing the embeddings of entities and relations randomly from uniform/Gaussian distributions [5]. For the training of the model, a batch of positive train triples S_m is fetched from the train triple set, \mathbb{S} . A negative sampling method is then used to generate a batch of negative triples. In the architecture, we inject our SNS method (shaded by yellowish color in Fig. 1) to generate the batch of negative triples, S'_m for the batch of positive triples, S_m . The batches of positive and negative triples are then used by the pairwise training strategy to learn the embeddings. In pairwise training, the model tries to assign more plausibility score to a positive triple than its corresponding negative triple.

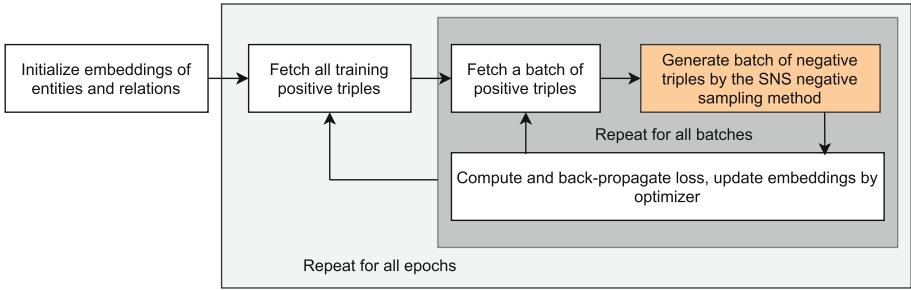


Fig. 1. Architecture of a classical KG embedding model with SNS sampling

The training objective is to optimize the embeddings of entities and relations for minimizing the total pairwise loss as designed Eq. 1.

$$\min_{\Theta} \sum_{\forall (h, r, t) \in S_m, (h', r, t') \in S'_m} L(f_r(h, t), f_r(h', t')) + \lambda reg(\Theta) \quad (1)$$

Here, f_r is the scoring function of the embedding model, $(h, r, t) \in S_m$ is a positive triple and $(h', r, t') \in S'_m$ is the corresponding negative triple. The pairwise loss for the positive and its negative triple is defined in Eq. 2 [5].

$$L(f_r(h, t), f_r(h', t')) = [\lambda - f_r(h, t) + f_r(h', t')]_+ \quad (2)$$

Here, λ is the margin and $[.]_+ = max(0, .)$ is the hinge function. The embedding updating process is repeated for all batches of positive triples (shaded by dark gray color in Fig. 1), and the whole training process (shaded by light gray color in Fig. 1) is repeated for T times (or epochs). The model training process is similar to a traditional KG embedding model except we adapt our negative sampling method. We refer to [5] for more details about the traditional KG embedding.

In the following, we describe our proposed SNS method for negative sample generation. The SNS method aims to generate high-quality negative triples for avoiding the 'vanishing-gradient' problem of uniform-random sampling, the complex parameter optimization problem of GAN-based sampling and the excessive memory requirement problems of NSCCaching. Figure 2a shows the basic steps of SNS sampling. The steps are described in the following.

Step 1. Triple perturbation: From Fig. 2a, SNS starts with generating a initial negative set for a positive triple by positive triple perturbation. This step is similar to other sampling methods. In triple perturbation, the head/tail of the positive triple is corrupted by replacing head/tail with other entities in the entity set (\mathbb{E}). In the same time, it is checked that the negative set does not contain any positive triple. To illustrate, consider the positive triple $q = (h, r, t) \in \mathbb{S}$. Corrupting the tail(t) gives the initial negative set $q'_0(t) = \{(h, r, t') \notin \mathbb{Q} | t' \in \mathbb{E}\}$.

Step 2. Candidate set generation: Generally, the size of the set $q'_0(t)$ is large as KG contains large number of entities. Zhang et al. [4] describe that only

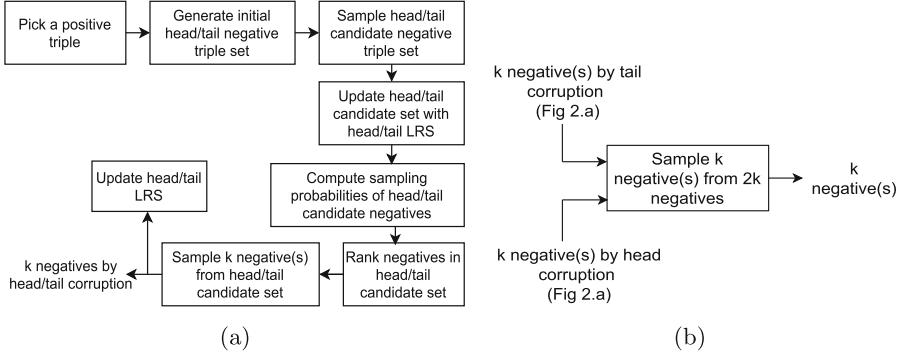


Fig. 2. The SNS sampling method (a) generation of k high-quality negatives by corrupting head/tail, (b) k negative(s) sampling from $2k$ high-quality negatives

some of initial negatives of the set are of good-quality. As we need few negatives for each positive triple, we randomly sample N_1 triples from $q'_0(t)$ to generate candidate negative set $q'_1(t)$ (i.e. $q'_1(t) \subseteq q'_0(t)$) for user defined parameter N_1 .

As the training progresses, it is desired that quality of the negative(s) for the next step will be better or close to the quality of negative(s) in the current step. For this purpose, the sampled negatives in the current step are stored in a small structure called least recently selected (LRS). $LRS[q'_h, q'_t]$ saves sampled negatives where q'_h and q'_t are the sampled negatives by head and tail corruption respectively in the previous step. The saved negatives are used in the next step for sampling high-quality negatives. The candidate negative set, $q'_1(t)$ is updated to include the LRS negative(s) as $q'_1(t) := q'_1(t) \cup LRS[q'_t]$. The use of LRS is intended to favour the exploitation behavior of the proposed SNS method. It ensures that the quality of the negative(s) at current step is better than or at least close to the quality of the negative(s) at the previous step.

Step 3. Sampling probability computation: In this step, we compute the sampling probability of each negative in the candidate negative set $q'_1(t)$. Negatives with higher probabilities are considered as high-quality negatives. The probability is defined based on the distance score of each negatives. The distance score for the negative triple, $(h, r, t'_i) \in q'_1(t)$ is computed as the distance between the corrupted(t) and the new entities(t') as Eq. 3.

$$d(t, t'_i) = \|\mathbf{t} - \mathbf{t}'_i\| \quad (3)$$

Here, \mathbf{t} and \mathbf{t}'_i are embeddings of the entities t and t'_i . A softmax function is then used to compute the sampling probability score of each candidate negative $(h, r, t'_i) \in q'_1(t)$ as Eq. 4.

$$P(h, r, t'_i) = \frac{\exp(\frac{1}{d(t, t'_i)})}{\sum_{j=1}^{N_1} \exp(\frac{1}{d(t, t'_j)})} \quad (4)$$

The softmax function computes higher sampling probability for the candidate negative triples with lower distance score.

Step 4. Negative triple sampling and LRS updating: The triples from the candidate negative set, $q'_1(t)$ are ranked in decreasing order of their probabilities and k negative(s) are sampled. A natural choice could be sampling top- k ($k=1$ for pairwise training, $k > 1$ for maximum likelihood training) negative(s). However, sampling top- k ranked negative(s) could arise two problems. Firstly, there is a chance of high repeat sampling of the same negative(s) (even in many consecutive steps) as the current candidate negative set also includes the least recently sampled (LRS) high-quality negative(s). This case affects the exploration of SNS sampling. Secondly, the existence of false negative triples (looks like high-quality) is not ignorable [5]. To tackle these problems, SNS randomly samples k negative triples from N_2 top ranked triples in q'_t as $q'_t = \{(h, r, t') | rank_{(h, r, t')} \leq N_2\}$ for user defined parameter N_2 where $N_2 > k$.

SNS repeats the above described process (from initial negative set generation to k negative(s) sampling) for head(h) corruption to sample k negative(s) as q'_h for the positive triple, q . The $LRS[q't, q'h]$ is updated with the sampled $2k$ high-quality negative(s). Finally, we randomly sample k high-quality negative(s) as $q'_{h,r,t}$ from $2k$ negatives in $q'_h \cup q'_t$. The sampled k negative(s), q' and the corresponding positive, q are then used to train the KG embedding model.

4 Experiments

To evaluate the efficiency of SNS sampling, we plug it to a KG embedding model. KG embedding models are broadly categorized into two main categories: (1) translational models and (2) semantic matching models [5]. Translational models consider each relation as a translation in the embedding space: adding relation to the head gives a close position to the tail [6]. On the other hand, semantic matching models define the plausibility of a triple by matching representations of entities and relations embodied in their embeddings [5]. The main focus of this paper is negative sampling for KGs. For evaluation, we choose TransH [11] to represent translational and DistMult [12] to represent semantic matching model as they are popular baselines for link prediction task in KGs. For details about the models, we refer to the original papers. The scoring functions of the models are given in Table 1. Each of the model is evaluated for three types of existing (i.e. random, GAN-based, NSCaching) and for the proposed SNS methods. We do not include SANS in the experiment due to its excessive memory requirement. For GAN-based method, we choose KBGAN as it is the only GAN-based sampling which has publicly available implementation (to best of our knowledge). We refer to the original articles for details about the sampling methods.

4.1 Datasets

In the experiments, we use five widely used benchmark KG datasets, i.e., FB15K, FB15K-237, WN18, WN18RR, YAGO3-10 for link prediction task [3, 4, 6]. WN18

Table 1. Scoring functions of KG embedding models: w_r is the normal vector of the hyperplane for the relation r , $diag(r)$ is the diagonal matrix for the relation r , and $\|\cdot\|_2$ represents l_2 norm.

Model	Embeddings	Scoring function, $f_r(h, t)$
TransH [11]	$\mathbf{h}^d, \mathbf{t}^d, \mathbf{r}^d, w_r^d$	$\ (\mathbf{h} - w_r^T \mathbf{h} w_r) + \mathbf{r} - (\mathbf{t} - w_r^T \mathbf{t} w_r)\ _2^2$
DistMult [12]	$\mathbf{h}^d, \mathbf{t}^d, \mathbf{r}^d$	$\mathbf{h}^T diag(r) \mathbf{t}$

Table 2. The experimental KG datasets

KG datasets	#Entity	#Relation type	#Facts	#Train	#Valid	#Test
WN18RR	40,943	11	93,003	86,835	3,034	3,134
WN18	40,943	18	151,442	141,442	5,000	5,000
FB15K-237	14,541	237	310,116	272,115	17,535	20,466
FB15K	14,951	1,345	592,213	483,142	50,000	59,071
FIGHT-HF-23R	90,430	23	948,298	853,482	47,402	47,414
YAGO3-10	113,273	37	1,089,040	1,079,040	5,000	5,000

is derived from the WordNet which is a large semantic lexicon for the English language. FB15K is a subset of triples from Freebase KG which is a large collaborative general knowledge base. WN18RR and FB15K-237 datasets are derived from WN18 and FB15K respectively after removing the inverse-duplicate relations. The YAGO3-10 dataset is extracted from the open source YAGO knowledge base considering the entities with at least 10 relations.

We also provide a biological KG dataset, namely FIGHT-HF-23R. FIGHT-HF is a biological knowledge graph which describes named relations among several biological types such as proteins, genes, diseases, drugs [13]. The original graph contains 246,672 biological entities and 24,601,110 relations of 37 types. We extract a medium size dataset by considering only those relations which have more than 50 facts and less than 500K facts. As a result, the dataset, naming FIGHT-HF-23R, contains 90,430 entities, 23 relation types and 948,298 triples. An example of extracted triple is (Cyclosporin A, drug_indication, Pterygium). The details about the dataset are available in GitLab repository¹.

The KG datasets, except the new FIGHT-HF-23R, come with train/valid/test splits. For FIGHT-HF-23R dataset, we split the dataset into train, validation, test triples. Unfortunately, we do not find any standard rule for splitting KG dataset. We split the dataset into 90/5/5 for train/validation/test triples so that we have enough triples to train the model. We apply the split rule to relation label where positive triples (facts) with a specific relation are also split with 90%/5%/5% ratio. This split ensures that we have all types of relations in all splits. We also check that no isolated entities exist in the training dataset as

¹ <https://gitlab.inria.fr/kislam/sns>.

in this case the entities will have low quality embedding. The split gives train, test and valid triple sizes as in Table 2.

4.2 Evaluation Metrics

To evaluate the performance of any sampling method, we plug it to a KG embedding model for link prediction task. The performance is defined with two widely used metrics: Hit@z, and mean reciprocal rank (MRR) [14]. The metrics are defined based on the rank of the positive test triple. Hit@z is defined as the average number of times a positive test triple is among the z highest ranked triples; whereas MRR is the average reciprocal rank of the positive test triple [15]. The range of both scores is 0 to 1. The higher value of MRR demonstrates the better ranking of positive test triples and better ranking provides better prediction performance. Also, higher Hit@z score indicates better performance. To illustrate, consider the positive test triple $q = (h, r, t) \in \mathbb{D}$. A set of negative triples $q'_t = \{(h, r, t') \notin \mathbb{Q} | t' \in \mathbb{E}\}$ is generated by simple triple perturbation (replacing tail with other entities) [6, 16] confirming that no positive triple exists in q'_t . The triples in $q \cup q'_t$ are then ranked in decreasing order of their scores (computed by embedding-based scoring functions in Table 1). The rank of the positive test triple q in $q \cup q'_t$ is defined as $rank_q^t$. Based on the rank of each positive test triple q, the performance metrics are defined in Eqs. 5 and 6.

$$Hit^t@z = \frac{1}{|\mathbb{D}|} \sum_{q \in \mathbb{D}} hit_q^t, \quad hit_q^t = \begin{cases} 1, & \text{if } rank_q^t \leq z \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$MRR^t = \frac{1}{|\mathbb{D}|} \sum_{q \in \mathbb{D}} \frac{1}{rank_q^t} \quad (6)$$

The whole evaluation process is also repeated by corrupting the head entity of each positive triple as well and $Hit^h@z$, MRR^h are computed. The final $Hit@z$, MRR metrics are the average of head and tails metrics i.e. $Hit@z = (Hit^t@z + Hit^h@z)/2$ and $MRR = (MRR^t + MRR^h)/2$. We re-scale the Hit@z score from the range 0–1 to 0–100 to facilitate the comparisons. As suggested by the most of the literature for link prediction in KGs, we consider $z \in \{1, 3, 10\}$.

4.3 Results and Discussion

The proposed SNS method is implemented in Python with well-known PyTorch and run on a ‘NVIDIA A100-PCIE-40GB’ GPU. We set training epoch number to 200, embedding dimension to 100, learning rate to 0.0001, margin value to 4.0 for all the experiments. For the embedding optimization task, we use the popular Adam optimizer [17]. We evaluate link prediction performance of embedding models with SNS sampling, and compare to other sampling methods in six KG datasets. Then, we do more analyses including parameter sensitivity,

performance in different epochs for the WN18RR dataset. The relation-wise prediction performance and examples of sampled negatives for WN18RR dataset are available in the same GitLab link as mentioned in Sect. 4.1.

Prediction Performance: We train the KG embedding models from scratch for each of the sampling methods, Random-uniform, KBGAN, NSCaching, SNS. For parameter setting, we follow the recommendations from the original paper ([4] for NSCaching, [3] for KBGAN). The prediction performance metrics are tabulated in Table 3. Considering translational model (TransH), undoubtedly the proposed SNS sampling method shows the best prediction metrics among all the negative sampling methods with respect to most of prediction metrics in most of the KG datasets. In WordNet KG datasets (WN18RR, WN18), NSCaching is the second best methods and the performance improved by 2–5% for SNS sampling considering Hit@k metrics when they are used with TransH. We find the best results of SNS sampling for FB15K dataset where the Hit@k scores are improved by 5–8% comparing to second best NSCaching sampling method. For rest of two datasets, SNS also remains best or second best in almost all metrics. Good balance between exploration and exploitation could be the most suitable reason behind this success. When the sampling methods are used in DistMult model, the hit@10 and hit@3 scores drop in all datasets except FIGHT-HF-23R. Training the model for more epochs might improve the metrics. However, our intention is not to compare different prediction models, rather different sampling methods. We see nearly similar trend of improvement in prediction performance for DistMult with SNS sampling. Surprisingly, the performance metric on the new FIGHT-HF-23R KG dataset are poor for TransH with the sampling methods, but better for DistMult with the sampling methods. In the following, we describe further analysis of prediction by TransH with different sampling methods in the smallest KG dataset, WN18RR.

Change in Prediction Performance for Different Epochs: To illustrate how the prediction performance changes with varying number of training epoch, we plot the MRR and Hit@10 scores of a embedding model with different sampling methods from epoch 10 to 200 with a interval of 10 in Fig. 3. At the initial point of epoch 10, the MRR score of SNS is nearly same (around 0.02) as other sampling methods, except KBGAN which has the highest MRR score (around 0.06). As the training epoch number increases, the embeddings quality are improved and consequently the rise in MRR scores of all methods are seen. The MRR improving rate is seen higher for SNS method. At the epoch 70, the MRR score of SNS is best. Though the improvement rate is not constant, the MRR scores of the proposed SNS method remain highest among all the sampling methods in the following epochs. We see the worst MRR for ‘uniform-random’ method as the method does not learn to pick high-quality negative triples. These improvements in rank are reflected in Hit@10 scores curves where SNS has the highest Hit@10 scores in later half of training epochs. These improvements in performance prove that our sampling method is able to provide better ranks of test triples than the state-of-art sampling methods.

Table 3. Link prediction (LP) results: MRR, and Hit@z of different negative sampling (NS) methods on KG datasets. The best and second best metrics are marked in bold and underline faces.

LP models	NS methods	WN18RR				WN18			
		MRR	hit@10	hit@3	hit@1	MRR	hit@10	hit@3	hit@1
TransH	Random	0.1520	32.27	23.72	0.11	0.3199	79.43	64.25	11.85
	NSCaching	<u>0.1713</u>	<u>40.68</u>	<u>31.43</u>	<u>0.93</u>	0.4171	<u>88.65</u>	<u>74.05</u>	17.48
	KBGAN	0.1708	40.08	29.35	0.10	<u>0.4183</u>	87.34	73.67	<u>18.09</u>
	SNS	0.1852	43.04	33.82	1.80	0.448	91.47	79.30	19.28
DistMult	Random	0.1918	32.16	23.88	14.22	0.3453	52.82	37.33	25.75
	NSCaching	0.2262	<u>37.37</u>	29.11	<u>17.84</u>	0.3772	<u>56.85</u>	42.18	<u>29.04</u>
	KBGAN	0.2285	33.42	<u>27.23</u>	17.34	<u>0.3791</u>	57.28	41.97	28.39
	SNS	0.2333	37.83	25.12	18.49	0.3931	56.46	<u>42.13</u>	31.38
FB15K237					FB15K				
TransH	Random	0.1988	36.68	22.50	11.50	0.3115	52.88	36.68	19.58
	NSCaching	0.2476	40.39	<u>26.59</u>	17.33	<u>0.3926</u>	61.22	<u>44.99</u>	<u>25.50</u>
	KBGAN	0.2162	<u>40.58</u>	23.52	<u>15.23</u>	0.3228	53.67	38.34	20.52
	SNS	0.2514	42.90	29.44	15.18	0.4360	66.72	51.52	30.58
DistMult	Random	0.1918	31.82	20.71	12.96	0.2188	33.25	21.84	12.95
	NSCaching	0.2205	34.87	25.69	<u>15.72</u>	<u>0.2327</u>	<u>39.89</u>	27.41	<u>16.19</u>
	KBGAN	0.2282	<u>36.23</u>	27.23	13.02	0.2203	35.54	23.12	15.92
	SNS	0.2471	38.18	26.97	17.80	0.2937	45.62	32.66	20.79
YAGO3-10					FIGHT-HF-23R				
TransH	Random	0.0850	18.96	9.05	1.24	0.0200	3.95	1.86	0.63
	NSCaching	0.1431	26.76	<u>15.97</u>	7.49	0.0294	6.96	1.97	0.66
	KBGAN	<u>0.1467</u>	26.08	<u>16.03</u>	<u>8.34</u>	0.0342	7.24	2.89	0.59
	SNS	0.1488	<u>26.72</u>	16.23	8.87	0.0410	8.92	<u>2.06</u>	0.64
DistMult	Random	0.0533	10.59	5.44	2.35	0.1439	26.01	14.29	8.75
	NSCaching	0.0875	<u>14.22</u>	8.86	5.64	0.1863	<u>30.36</u>	<u>20.13</u>	<u>13.10</u>
	KBGAN	0.0712	13.98	7.79	3.19	0.1681	28.25	18.64	14.38
	SNS	0.0804	16.72	<u>8.39</u>	<u>4.98</u>	0.1899	33.78	22.19	12.63

Parameter Sensitivity Analysis: SNS sampling method has two parameters, N_1 and N_2 . To describe the changes in performance for different values of these parameters, we record Hit@10 and MRR scores for different values of N_1 with fixed $N_2 = 5$ which are plotted in Fig. 4. As the value of N_1 increases, more initial negative triples are explored and the SNS sampling gets better exploration. As a consequence, the prediction performance improves as the value of N_1 increases from 20 to upper as seen from Figs. 4a, 4b. The prediction performance is nearly stable for $N_1 = 50$ and above. In the point $N_1 = 50$, SNS sampling has good exploration to sample sufficient number of high-quality negatives. And this could be the cause of performance stability for $N_1 \geq 50$. Again, to describe the sensitivity of the parameter N_2 among $\{1, 2, 3, 4, 5, 6, 7\}$ with fixed $N_1 = 50$ in Fig. 5. Figures 5a and 5b show the change in Hit@10 and MRR for change in N_2 . With

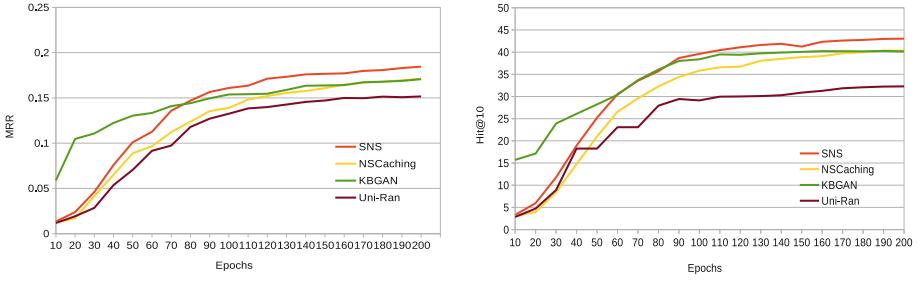


Fig. 3. Prediction scores of TransH with different samplings in different epochs

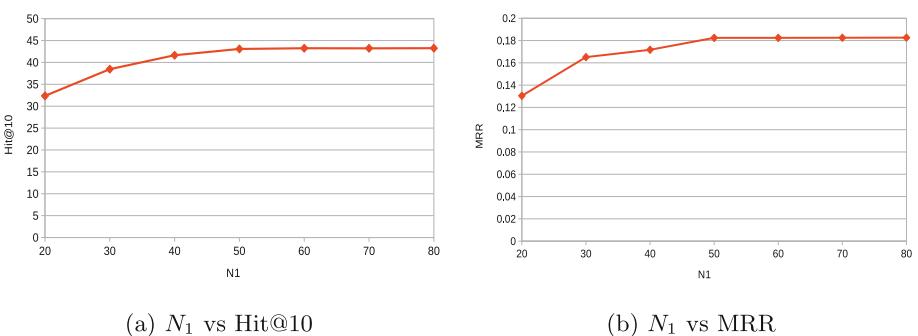
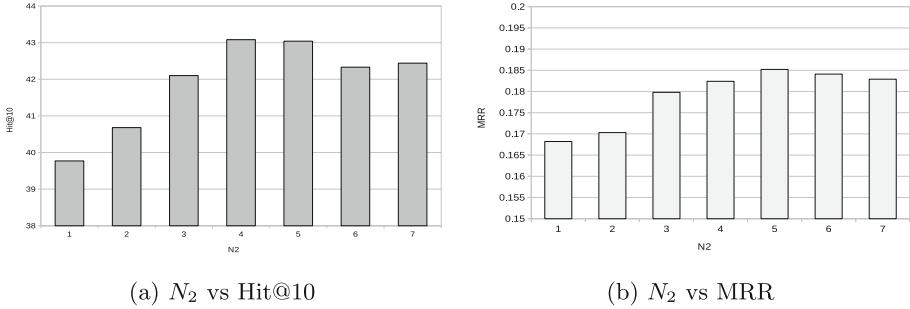


Fig. 4. Sensitivity of SNS to N_1 , size of candidate negative set

setting $N_2 = 1$, SNS samples the top-most ranked negative(s). In this case, the chance of repeat sampling is high as SNS considers already known high-quality LRS negative(s) in addition to other candidate negatives. In case of high repeat sampling, SNS suffers from low exploration and high exploitation effect leading to drops in MRR and Hit@10 metrics. With $N_2 = 1$, we see lowest prediction performance for SNS. As the value of N_2 increases, SNS gets better exploration and the best balance between exploration and exploitation is found for $N_2 = 5$ where the highest prediction metrics are recorded. However as the value of N_2 increases, the chance of sampling of known high-quality triple decreases (poor exploitation) and the chance of sampling less good-quality negative increases. As a result, the performance drops as seen in Fig. 5 where both hit@10 and MRR drop for $N_2 > 5$.

Memory and Computational Efficiency. Undoubtedly, random-uniform is the simplest, fastest, memory efficient sampling method as it does not learn or store any parameter. GAN-based sampling makes the prediction model more complex, increases the number of training parameters, and makes the model harder to train due to use of reinforcement learning [4]. As a consequence, KBGAN needs extra memory and computational cost to store and optimize

**Fig. 5.** Sensitivity of N_2

the parameters. NSCaching stores set of high-quality negative triples in each positive triple cache which makes it worst memory efficient. In addition, the method takes additional time to update the cache periodically. The proposed SNS sampling method does not increase the training parameter like GAN-based method. It memorizes only the least recently sampled negative triple which takes very small amount of memory. Thus, intuitively SNS sampling is more memory efficient than GAN-based sampling and NSCaching. The method does not use complex learning method like GAN-based method or does not take extra cache updating time like NSCaching. It takes very small amount of time to update the LRS structure. We record the training time of each sampling method with the TransH embedding model. The data on training time is available at in the same GitLab link as mentioned in Sect. 4.1. We see that the training time increases as the number of training samples increases, as expected. We find 7–40% and 3–14% improvement in training time for SNS when it is compared to KBGAN and NSCaching respectively.

5 Conclusion

In this paper, we propose a simple and efficient negative sampling method for knowledge graph embedding. The method is general and can be plugged to any knowledge graph embedding method. The method is able to generate high-quality negative triples and takes low computational time and memory while anticipating the ‘vanishing-gradient’ problem. Experimentally, we evaluate our method on six knowledge graph datasets for link prediction task and also describe its parameter sensitivity. The results show that the proposed SNS sampling brings consistent improvements in prediction performance.

The poor performance of the studied models on YAGO3-10 and FIGHT-HF-23R datasets leaves the future work to explore other embedding models in the literature on these datasets. Implementing the models in a distributed environment to improve computational efficiency is another potential perspective of this work.

References

1. Paulheim, H.: Knowledge graph refinement: a survey of approaches and evaluation methods. *Semant. Web* **8**(3), 489–508 (2017)
2. Dong, X., et al.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 601–610 (2014)
3. Cai, L., Wang, W.Y.: KBGAN: adversarial learning for knowledge graph embeddings. In: 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1470–1480 (2018)
4. Zhang, Y., Yao, Q., Shao, Y., Chen, L.: NSCaching: simple and efficient negative sampling for knowledge graph embedding. In: 2019 IEEE 35th International Conference on Data Engineering, pp. 614–625 (2019)
5. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **29**(12), 2724–2743 (2017)
6. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, vol. 26, pp. 1–9 (2013)
7. Hu, K., Liu, H., Hao, T.: A knowledge selective adversarial network for link prediction in knowledge graph. In: Tang, J., Kan, M.Y., Zhao, D., Li, S., Zan, H. (eds.) CCF International Conference on Natural Language Processing and Chinese Computing. LNCS, vol. 11838, pp. 171–183. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32233-5_14
8. Wang, P., Li, S., Pan, R.: Incorporating GAN for negative sampling in knowledge representation learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
9. Ahrabian, K., Feizi, A., Salehi, Y., Hamilton, W.L., Bose, A.J.: Structure aware negative sampling in knowledge graphs. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 6093–6101 (2020)
10. Chen, J., Xin, B., Peng, Z., Dou, L., Zhang, J.: Optimal contraction theorem for exploration-exploitation tradeoff in search and optimization. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **39**(3), 680–691 (2009)
11. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28, no. 1, pp. 1112–1119 (2014)
12. Yang, B., Yih, W.T., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: International Conference Learning Representation (2014)
13. Bresso, E., et al.: A data science approach for exploring differential expression profiles of genes in transcriptomic studies-application to the understanding of ageing in obese and lean rats in the FIGHT-HF project. In: JOBIM 2018-Journées Ouvertes Biologie, Informatique et Mathématiques (2018)
14. Wang, M., Qiu, L., Wang, X.: A survey on knowledge graph embeddings for link prediction. *Symmetry* **13**(3), 485 (2021)
15. Rossi, A., Matinata, A.: Knowledge graph embeddings: are relation-learning models learning relations? In: EDBT/ICDT Workshops (2020)

16. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol. 1, pp. 687–696 (2015)
17. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representation (2015)



Evaluation of Anomaly Detection for Cybersecurity Using Inductive Node Embedding with Convolutional Graph Neural Networks

Amani Abou Rida^{1(✉)}, Rabih Amhaz^{1,2}, and Pierre Parrend^{1,3}

¹ Université de Strasbourg, CNRS, ICube (Laboratoire des sciences de l'ingénieur, de l'informatique et de l'imagerie), 67000, UMR 7357 Strasbourg, France

abou-rida.amani@etu.unistra.fr, {amhaz,parrend}@unistra.fr

² ECAM Strasbourg-Europe, 67300 Schiltigheim, France

³ EPITA, 5, Rue Gustave Adolphe Hirn, 67000 Strasbourg, France

Abstract. In the face of continuous cyberattacks, many scientists have proposed machine learning-based network anomaly detection methods. While deep learning effectively captures unseen patterns of Euclidean data, there is a huge number of applications where data are described in the form of graphs. Graph analysis have improved detecting anomalies in non-Euclidean domains, but it suffered from high computational cost. Graph embeddings have solved this problem by converting each node in the network into low dimensional representation, but it lacks the ability to generalize to unseen nodes. Graph convolution neural network methods solve this problem through inductive node embedding (inductive GNN). Inductive GNN shows better performance in detecting anomalies with less complexity than graph analysis and graph embedding methods.

Keywords: Anomaly detection · Convolutional graph neural network · Inductive graph learning · Graph embedding · Graph analysis · Link prediction

1 Introduction

Anomaly detection is any method for finding events that don't match a given expectation. In the face of continuous cyberattacks, many scientist have proposed machine learning-based network anomaly detection methods [1] such as one-class support vector machines (OSVM), autoencoders (AE), and isolation forests (IS). These methods have proven to be very effective in detecting anomalies [2] and they discover hidden patterns in Euclidean data [3] where they are being plotted in n-dimensional linear space. However, there is an increasing number of applications where data are expressed in the form of graphs [4]. Unlike the n-dimensional grid-like Euclidean space data (images, audio and text), network data represent irregular non-Euclidean domains [5]. A graph can be used as an effective tool to describe and model the complex structure of network data. A graph $G(V, E)$ is

typically defined as a set of vertices indicated by V , and edges indicated by E between different vertices [4]. Graph-structured data are used to model complex systems, ranging from social media networks [6], traffic networks [7] to financial nets [8]. Consequently, detecting anomalies from graphs has become an important research problem [9]. We obtain rich information from the graph relation structures which provide a natural way to understand links between entities [5]. Graph analysis [4] is used for advanced quantitative characterisation and control of large-scale networks, but traditional methods suffer from high computational cost and excessive memory requirements. Graph embedding [4] methods are effective in transforming high-dimensional graphs into low-dimensional representation of dense and continuous vector spaces, keeping the graph structure properties. However, generating new representations using graph embedding at each analysis and updating all node embeddings is costly. Therefore, scientists proposed to update the embeddings according to the changes incrementally instead of relearning the whole embeddings again. Graph convolutional neural networks can address the graph embedding problem by dealing with graphs incrementally using an inductive node embedding model [10]. This incremental computation ensures its efficiency since each new node only needs to sample and aggregate its neighbor's features. The main objective of this research is to further investigate an anomaly detection system that is suitable for detecting anomalies incrementally having the best performance. To accomplish this objective, the research addresses the following question:

- How can inductive node embedding graph convolutional neural networks (in short Inductive GNNs) improve the prediction performance in anomaly detection over the traditional graph analysis and graph embedding methods?

The paper is organised as follows. Section 2 introduces the state of the art, Sect. 3 identifies the requirements for the schemes under evaluation. Section 4 defines the process for anomaly detection using Inductive GNNs, Sect. 5 presents its implementation. Section 6 provides the evaluations and Sect. 7 discusses the implications of these evaluation. Section 8 concludes this work.

2 State of the Art

In this section we present the use of graph analysis and embedding for detecting anomalies. We discuss the benefits and limits of the methods, and how Inductive Node Embedding Graph Convolutional Neural Networks can address them.

Graph analysis [4] is a process for analyzing data in graph structures, in a **Non-Euclidean Space** using data points as nodes and relationships as edges. There is an increasing number of applications where data are represented as a graph with complex relationships and inter-dependency between objects. In other words, analyses are increasingly going from Euclidean (e.g., images, audio and text) to non-Euclidean space with higher number of dimensions. However, to perform analysis on non-Euclidean space, graph analysis methods have came into that can help improving the quantitative understanding and the control of

complex networks. Graph analysis methods, such as connectivity analysis, community detection analysis and centrality analysis, are largely based on extracting handcrafted graph topological features of nodes and edges directly from the adjacency matrices. In our work, we use the Speaker-Listener Label Propagation Algorithm (SLLPA) [11]. SLLPA is an improvement of the Label Propagation algorithm that is capable of detecting multiple communities per node. When applied to large-scale network analysis, these methods may suffer from high computational cost and excessive memory requirements as a result of high-dimensionality [12]. In addition, hand-engineered features are often task-specific and cannot bring identical performance when re-used for other tasks [4].

Graph embedding [4] techniques have shown an important role for the capacity of transforming high-dimensional sparse graphs into **low-dimensional representations**, dense and continuous vector spaces. The main purpose of graph embedding methods is to encode nodes into a latent vector space and to pack every node's properties into a vector with a lower dimension. Graph embedding methods integrate three complementary domains [13]: matrix factorization, random walk, and neural network methods [13]. In this paper we focus on Random walk-based methods, in particular Node2Vec [14] and fast random projection (FastRP) [15]. Node2Vec is a node embedding method that measures a vector illustration of a node where the neighborhood is sampled using random walks [14]. FastRP is a scalable and performant algorithm for preserving similarity between nodes and their neighbors. This means that two nodes that have similar neighborhoods should be assigned similar embedding vectors [15]. The limitations of graph embedding methods are:

1. They cannot easily scale up to large network embeddings and only consider local connections.
2. The model should be able to generate embeddings for some target nodes as soon as new information has been made available.
3. The generation of a representation using graph embedding at each time and updating all the node embeddings is costly.

Graph Neural Networks (GNNs) [5] are a significant stride to operate precisely on graph-structured data, and a promising method for solving above limitations. The features of the neighbours of a graph node are aggregated and passed as a message to that node. After different aggregation iterations, the feature vector of a node extracts the structural information from the node's neighborhood [9]. The final result, i.e., the aggregated information obtained at each node, is referred to as the node embedding. GNNs have non-linear activation functions and parallelization skills, which can solve the data non-linearity and the computational complexity problems, respectively [16]. Graph neural networks (GNNs) are classified into 4 categories: recurrent graph neural networks (RecGNNs), convolutional graph neural networks (ConvGNNs), graph autoencoders (GAEs), and spatial-temporal graph neural networks (STGNNs) [5]. Although GNNs have established outstanding performance in many graph mining tasks [6, 7, 17], it remains unclear how to accomplish their potentiality for Graph Anomaly Detection [16].

Convolutional Graph Neural Networks (ConvGNNs) acquire the movement of convolution from grid data (Euclidean structure) to graph data (non-Euclidean structure) [5]. ConvGNNs play an important role in building up many other complex GNN models [5]. They fall into two categories, spectral-based and spatial-based. Spectral based approaches specify graph convolutions by proposing filters from graph signal processing [18]. Spatial-based approaches perform graph convolutions locally on each node where weights can be easily shared across different locations and structures [19]. In recent years, some ConvGNNs methods for learning over graphs have been proposed. These methods do not scale to large graphs or are designed for whole-graph classification [20–22]. However, an inductive node embedding method is needed to adopt sampling so as to provide a fix number of neighbors for each node [22]. In the next section we introduce and define Inductive GNNs. Inductive GNNs are an inductive way to iteratively update the node embeddings. It follows an embedding propagation schema where the embedding of a node is recursively updated by aggregating values propagated from its neighboring nodes. Inductive GNN apply node feature information to achieve node embeddings on unseen nodes or graphs [23]. Rather than training individual embeddings for every node, the algorithm learns a function that achieves embeddings by sampling and aggregating features from a node's regional neighborhood [22]. This means that inductive GNNs are able to update the embeddings according to the changes incrementally instead of relearning the embeddings whole again. It performs graph convolutions according to (1):

$$h_v^{(k)} = \sigma(W^{(k)} \cdot f_k(h_v^{(k-1)}, \{h_u^{(k-1)}, \forall u \in S_N(v)\})) \quad (1)$$

where $h_v^{(0)} = x_v$, $f_k(\cdot)$ is an aggregation function, $S_N(v)$ is a random sample of the node v's neighbors [5].

3 Requirements

The first step for identifying the right model to detect anomalies using Inductive GNNs is to define the requirements of such a model. The model aims at detecting anomalies in graphs using graph convolutional neural network methods. These requirements are derived from the literature [5, 10, 22, 24, 25] and from our experience, Detecting anomalies using Inductive GNNs model should have the following requirements:

- *Transferability*: The nodes embeddings can be fed into downstream Machine Learning methods [24].
- *Scalability*: Generation of the embeddings should be as fast and scalable as possible. This is measured according to two properties: 1) Dimensionality [5]: Dimensions of the embedding play a fundamental role in the selection of the approach to be applied. 2) Inductive embedding generation [22]: Iterative generation of node embedding as soon as new information have been made available without retraining the node embedding of all the graph again.

4 Anomaly Detection Using Inductive GNNs

In this section we see how to use Inductive GNNs for graph anomaly detection. In particular, Inductive GNNs based methods for detecting anomalies can update the embeddings according to the changes incrementally instead of relearning the embeddings whole again. Moreover, we compare Inductive GNNs with graph analysis and graph embedding algorithms such as FastRP and Node2vec to evaluate their relative anomaly detection capabilities.

The steps needed to detect anomalies according to the above requirements are the following ones:

1. Load a dataset as a csv, json, or pcap extension inside the graph database. The standardized data format of the dataset should include the following properties: IP address, start timestamp, end timestamp, PKseqID, and a label attack to determine if this node is normal or anomalous. These properties represent the node in the graph.
2. Create the graph composed of nodes and edges. In our work we represent a node as an event. The edges represent a relationship between two events and they are constructed according to the following conditions:
 - (a) Events should have the same source IP address.
 - (b) The difference between the end timestamp and start timestamp between each event should be less than 20 s.

When these two conditions are satisfied between two different events a relationship “Connected-To” is created as an edge between them.

3. Apply different Graph Data Science methods on the graph.
 - (a) Apply Graph Analysis: community detection, centrality, and similarity algorithms.
 - (b) Apply Graph embedding algorithms: Node2Vec and FastRP.
 - (c) Apply Graph Convolutional Neural Network: Inductive GNNs.
4. To detect anomalies in the graph, apply the output of Inductive GNNs on the output of Graph Analysis and Graph Embedding to compare the results. Anomaly detection should there exhibit better performance and more accurate results than the traditional methods.

Our model consists of three parts, as shown in Fig. 1: Input, process, and output. The input of our model is the dataset that is converted into graph and stored in a graph database. The process consists in applying graph analysis, graph embedding and Inductive GNNs methods on the graph. The output shows that the performance of detecting anomalies using Inductive GNNs on graph analysis and graph embedding is improved according to the performance and accuracy.

5 Implementation

We consider 2 datasets for the evaluation: BoT-IoT [26] and UNSW-NB15 [27]. They are created by designing a realistic network environment in the Cyber Range Lab of UNSW Canberra. In order to have some statistics on these datasets we implemented “Data Summary” (`data_sum`), a Python application

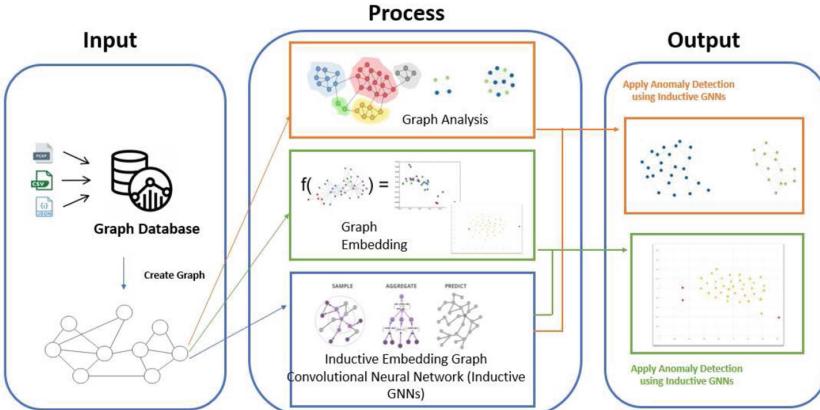


Fig. 1. Process for detecting anomalies using Inductive GNNs

that extracts the total number of records and the number of attack records inside a dataset. The number of records of BoT-IoT is 72,388,626 and the number of attack records is 30,668,045. The types of attacks are: DDoS, DoS, OS and Service Scan, Keylogging and Data exfiltration. The number of records of UNSW-NB15 dataset is 540,044 and the number of attack records is 164,673. The types of attacks are: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms.

Neo4j¹ is an in memory graph handling environment that offers an integrated graph database for persistence so there is no need to recreate the graph each time it changes [28]. Moreover, Neo4j offers a graph data science library that can be used for our process to apply graph analysis, graph embedding, and Inductive GNNs methods, in particular the GraphSAGE library which is the reference implementation. Detecting anomalies using GraphSAGE is performed through cypher graph query language [29] which is used to query the Neo4j graph database for creating, storing, performing graph data science libraries.

The hyperparameter used in Neo4J GraphSAGE implementation are: dimension $d = 64$, sample normalization vector $S = [25,10]$, and aggregator function. GraphSAGE provides in particular GraphSAGE-Mean and GraphSAGE-Pool aggregation strategies. The mean operator aggregates the neighbours' vectors by computing their element-wise mean. The pooling aggregator, instead, uses the neighbours' vectors as input to a fully connected layer before performing the concatenation, and then it applies elementwise max-pooling operation.

6 Evaluation

In this section, we perform a set of experiments to evaluate the performance of anomaly detection using Inductive GNNs model compared to different graph

¹ <https://neo4j.com/>.

analysis and graph embedding algorithms. The evaluation of anomaly detection is performed for the two core target properties: Transferability, and Scalability.

To evaluate the **Transferability** of detecting anomalies using Inductive GNNs, we challenge the model for its capability to be fed into downstream Machine Learning applications. We apply the K-Nearest Neighbors (KNN) algorithm on the output training of GraphSAGE. KNN computes a distance value for all node pairs in the graph and creates new relationships between each node and its k nearest neighbors [30]. This operation leads to a new relationship called “SIMILAR-GraphSAGE”. This relationship is used as an output of GraphSAGE and it is used for comparing the results of using SLLPA (graph analysis method) and FastRP (graph embedding) to SLLPA-GraphSAGE and FastRP-GraphSAGE methods as shown in Figs. 2 and 3. First we compare GraphSAGE to graph analysis (SLLPA method). Then we compare GraphSAGE to graph embedding (FastRP method). SLLPA is used to cluster the graphs according to attack and normal events. Figures 2a and 2b shows the clustering using SLLPA method and clustering using SLLPA on GraphSAGE output respectively. Label 0 on the event means that the event is normal (blue) and label 1 means that the event is an attack (green). The result in Fig. 2a shows two clusters containing 6 attack events (2 in the first cluster and 4 in the second one), while the results in Fig. 2b shows two clusters containing 14 attack events all in the same cluster. Moreover, Figs. 3a and 3b show the embedding using FastRP method and embedding using FastRP on GraphSAGE output respectively. The red events in the figures are the attack events and the yellow events are the normal ones. The result in Fig. 3a show the detection of 7 attack event, while the result in Fig. 3b show the detection of 9 attack events.

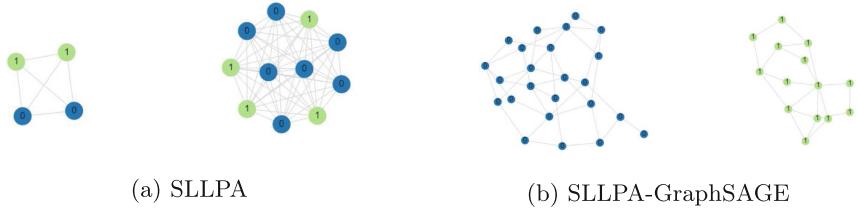


Fig. 2. Comparing the dynamicity of graph analysis (SLLPA) to graph neural network (GraphSAGE implementation)

The generation of the embeddings should be as fast and scalable as possible. **Scalability** is measured according to two main properties: 1) Inductive embedding generation [22] and 1) Embedding Dimensions [5]. The model should be able to generate embeddings in an iterative way as soon as new information has been made available without re-running the embedding all over again. To evaluate the scalability according to inductivity we use Link prediction method to compute the time and F1 score for graph analysis, graph embedding, and graph convolutional neural network methods. Link prediction is a common machine

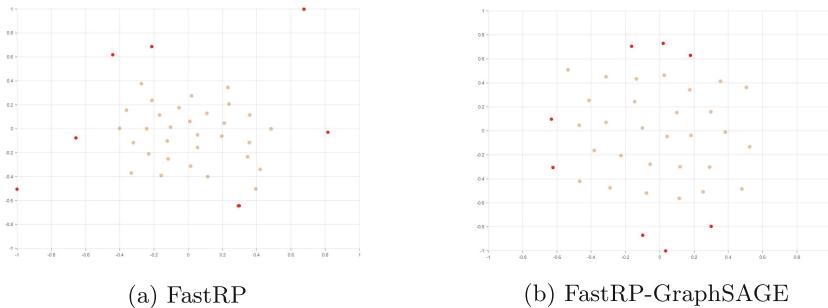
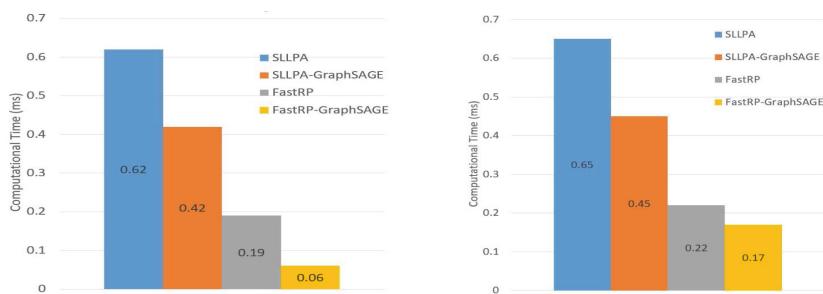


Fig. 3. Comparing the dynamicity of graph embedding (FastRP) to graph neural network (GraphSAGE implementation)



(a) Computational time for graph analysis (SLLPA, SLLPA-GraphSAGE) and graph embedding (FastRP, FastRP-GraphSAGE) for BoT-IoT dataset (b) Computational time for graph analysis (SLLPA, SLLPA-GraphSAGE) and graph embedding (FastRP, FastRP-GraphSAGE) for UNSW-NB15 dataset

Fig. 4. Computational time for graph analysis (SLLPA, SLLPA-GraphSAGE) and graph embedding (FastRP, FastRP-GraphSAGE) for BoT-IoT and UNSW-NB15 datasets

learning task applied to graphs [31]: training a model to learn, between pairs of nodes in a graph, where relationships should exist. Moreover, this method is used to calculate the precision, recall, and F1-Score. The F1-Score is the harmonic mean of Precision and Recall, this score is widely exploited since it is a trade-off among the previous metrics and consent to have a better understanding of the predictive performance of the model. Table 1 and Table 2 show the value of the precision, recall, F1-Score, and duration of time for applying link prediction on SLLPA, Node2Vec, FastRP, GraphSAGE-Mean, and GraphSAGE-Pool for BoT-IoT and UNSW-NB15 respectively. GraphSAGE-Pool has the highest F1-score 0.807, 0.751, and the less time duration 0.15, 0.08 ms in both BoT-IoT and UNSW-NB15 datasets respectively.

Dimensions of the embedding play a fundamental role in the selection of the approach to be applied. To evaluate the Scalability according to dimensionality we use different embedding dimensions. The embedding dimensions represent

Table 1. Link prediction performance on SLLPA, Node2Vec, FastRP, GraphSAGE-Mean, and GraphSAGE-Pool for BoT-IoT dataset

Link prediction	Precision	Recall	F1-score	Time duration
SLLPA	0.748	0.757	0.752	0.77
Node2Vec	0.742	0.757	0.749	0.54
FastRP	0.749	0.755	0.752	0.54
GraphSAGE-Mean	0.755	0.76	0.757	0.22
GraphSAGE-Pool	0.798	0.817	0.807	0.15

Table 2. Link prediction performance on SLLPA, Node2Vec, FastRP, GraphSAGE-Mean, and GraphSAGE-Pool for UNSW-NB15 dataset

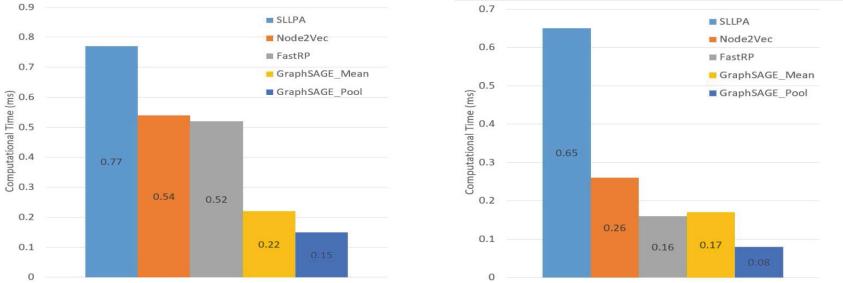
Link prediction	Precision	Recall	F1-score	Time duration
SLLPA	0.743	0.738	0.74	0.65
Node2Vec	0.742	0.751	0.7464	0.26
FastRP	0.751	0.743	0.7469	0.16
GraphSAGE-Mean	0.7516	0.745	0.748	0.17
GraphSAGE-Pool	0.7516	0.7517	0.751	0.08

the dimension of the generated node embeddings as well as their hidden layer representations. We have chosen the following dimensions $d = 1$, $d = 10$, $d = 20$, $d = 40$, $d = 64$, and $d = 128$. Figures 5c and 5d shows the computational time with respect to different embedding dimensions for Node2Vec, FastRP, GraphSAGE-Mean, and GraphSAGE-Pool. GraphSAGE-Pool has the best time 0.04 ms and 0.02 ms in both BoT-IoT and UNSW-NB15 datasets respectively.

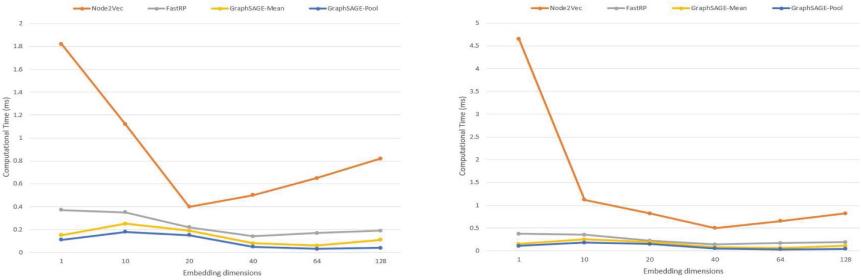
7 Discussion

The evaluation of **Transferability** shows that using Inductive GNNs we can get more anomalies detected. The accuracy of clustering is also being improved, since we are getting one cluster for the normal events and one different cluster for the attack events. Moreover, although using SLLPA-GraphSAGE method is detecting more anomalies than FastRP-GraphSAGE, Figs. 4a and 4b show that computational time for FastRP-GraphSAGE is 0.06 ms, 0.17 ms less than the computational time using SLLPA-GraphSAGE 0.42 ms, 0.45 ms in both BoT-IoT and UNSW-NB15 datasets respectively. Thus using inductive Node Embedding to detect anomalies have less complexity than using graph analysis.

The evaluation of **Scalability** from the Table 1 and Table 2 shows that GraphSAGE-Pool is preferable since it has the higher F1-Score with less computational time than the other methods. Figures 5a and 5b show that Link prediction on GraphSAGE-Pool provides the best complexity score in terms of computational time. Thus having new information in the graph gives more accurate



(a) Computational time for link prediction on SLLPA, Node2Vec, FastRP, GraphSAGE-Mean, and GraphSAGE-Pool for BoT-IoT dataset
(b) Computational time for link prediction on SLLPA, Node2Vec, FastRP, GraphSAGE-Mean, and GraphSAGE-Pool for UNSW-NB15 dataset



(c) Variation of time with respect to different dimensions for Node2Vec, FastRP, GraphSAGE-Mean and GraphSAGE-Pool for BoT-ToT dataset
(d) Variation of time with respect to different dimensions for Node2Vec, FastRP, GraphSAGE-Mean and GraphSAGE-Pool for UNSW-NB15 dataset

Fig. 5. Scalability of GraphSAGE (mean and pool) compared to SLLPA, Node2Vec, and FastRP

result and better performance when using GraphSAGE-Pool. The evaluation of scalability according to different dimensions shows that having a greater dimension offers a greater precision, but is more costly to operate over. However, when comparing the complexity of Node2Vec and FastRP to Inductive GNNs one can notice that Inductive GNNs is more scalable and faster. GraphSAGE-Pool shows some slight improvement in its performance compared to GraphSAGE-Mean.

8 Conclusions and Perspectives

Detecting anomalies using inductive Node Embedding with Convolutional Graph Neural Networks proves to comply with the requirements for scalability and transferability. Experiments shows that inductive embedding graph convolutional neural networks improve the performance of detecting anomalies compared to graph analysis and graph embedding. This proposal opens a great challenge for the capability of detecting anomalies on heterogeneous and dynamic graphs.

Heterogeneous graphs have many different types of vertices and many types of edges, and they make the process of calculating embeddings more complicated. Dynamic graphs are the graphs where their nodes/edges may change over time. For this, we should propose new Inductive Node Embedding GNN that is able to detect anomalies on heterogeneous and dynamic graphs.

References

1. Xiao, Q., Liu, J., Wang, Q., Jiang, Z., Wang, X., Yao, Y.: Towards network anomaly detection using graph embedding. In: Krzhizhanovskaya, V.V., et al. (eds.) ICCS 2020. LNCS, vol. 12140, pp. 156–169. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50423-6_12
2. Tonejc, J., Güttes, S., Kobekova, A., Kaur, J.: Machine learning methods for anomaly detection in Bacnet networks. *J. Univ. Comput. Sci.* **22**(9), 1203–1224 (2016)
3. Barrett, P.: Euclidean distance: raw, normalised, and double-scaled coefficients. *Tech. Rep.* (2005)
4. Xu, M.: Understanding graph embedding methods and their applications. arXiv preprint. [arXiv:2012.08019](https://arxiv.org/abs/2012.08019) (2020)
5. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **32**(1), 4–24 (2020)
6. Peng, H., et al.: Fine-grained event categorization with heterogeneous graph convolutional networks. arXiv preprint. [arXiv:1906.04580](https://arxiv.org/abs/1906.04580) (2019)
7. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. arXiv preprint [arXiv:1709.04875](https://arxiv.org/abs/1709.04875) (2017)
8. Weber, M.: Scalable graph learning for anti-money laundering: A first look. arXiv preprint. [arXiv:1812.00076](https://arxiv.org/abs/1812.00076) (2018)
9. Wang, X., Du, Y., Cui, P., Yang, Y.: OCGNN: one-class classification with graph neural networks. arXiv preprint. [arXiv:2002.09594](https://arxiv.org/abs/2002.09594) (2020)
10. Cui, Z., et al.: DynGCN: dynamic graph embedding with graph convolutional network. arXiv preprint. [arXiv:2104.02962](https://arxiv.org/abs/2104.02962) (2021)
11. Xie, J., Szymanski, B.K., Liu, X.: Slpa: uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In: 2011 IEEE 11th International Conference on Data Mining Workshops, pp. 344–349. IEEE (2011)
12. Su, C., Tong, J., Zhu, Y., Cui, P., Wang, F.: Network embedding in biomedical data science. *Brief. Bioinform.* **21**(1), 182–197 (2020)
13. Cui, P., Wang, X., Pei, J., Zhu, W.: A survey on network embedding. *IEEE Trans. Knowl. Data Eng.* **31**(5), 833–852 (2018)
14. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864 (2016)
15. Chen, H., Sultan, S.F., Tian, Y., Chen, M., Skiena, S.: Fast and accurate network embeddings via very sparse random projection. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 399–408 (2019)
16. Wang, X., Jin, B., Du, Y., Cui, P., Yang, Y.: One-class graph neural networks for anomaly detection in attributed networks. arXiv preprint [arXiv:2002.09594](https://arxiv.org/abs/2002.09594) (2020)

17. Wang, D., et al.: A semi-supervised graph attentive network for financial fraud detection. In: 2019 IEEE International Conference on Data Mining (ICDM), pp. 598–607. IEEE (2019)
18. Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P.: The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. *IEEE Sig. Process. Mag.* **30**(3), 83–98 (2013)
19. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 922–929 (2019)
20. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. arXiv preprint. [arXiv:1312.6203](https://arxiv.org/abs/1312.6203) (2013)
21. Duvenaud, D., et al.: Convolutional networks on graphs for learning molecular fingerprints. arXiv preprint arXiv:1509.09292 (2015)
22. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 1025–1035 (2017)
23. Pande, A., Ni, K., Kini, V.: Swag: item recommendations using convolutions on weighted graphs. In: 2019 IEEE International Conference on Big Data (Big Data), pp. 2903–2912, IEEE (2019)
24. Martignano, A.: Real-time anomaly detection on financial data (2020)
25. Yu, D., Yang, Y., Zhang, R., Wu, Y.: Knowledge embedding based graph convolutional network. In: Proceedings of the Web Conference, 2021, pp. 1619–1628 (2021)
26. Koroniots, N., Moustafa, N., Sitnikova, E., Turnbull, B.: Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. *Fut. Gen. Comput. Syst.* **100**, 779–796 (2019)
27. Moustafa, N., Slay, J.: The evaluation of network anomaly detection systems: statistical analysis of the UNSW-nb15 data set and the comparison with the kdd99 data set. *Inf. Secur. J. Glob. Perspect.* **25**(1–3), 18–31 (2016)
28. Akoglu, L., Faloutsos, C.: Anomaly, event, and fraud detection in large network datasets. In: Proceedings of the sixth ACM International Conference on Web Search and Data Mining, pp. 773–774 (2013)
29. Holzschuher, F., Peinl, R.: Performance of graph query languages: comparison of cypher, gremlin and native access in neo4j. In: Proceedings of the Joint EDBT/ICDT 2013 Workshops, pp. 195–204 (2013)
30. Dong, W., Moses, C., Li, K.: Efficient k-nearest neighbor graph construction for generic similarity measures. In: Proceedings of the 20th International Conference on World Wide Web, pp. 577–586 (2011)
31. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. *J. Am. Soc. Inf. Technol.* **58**(7), 1019–1031 (2007)



Hybrid Graph Embedding Techniques in Estimated Time of Arrival Task

Vadim Porvatov^{1,2(✉)}, Natalia Semenova¹, and Andrey Chertok^{1,3}

¹ Sberbank, Moscow 117997, Russia

achertok@sberbank.ru

² National University of Science and Technology “MISIS”, Moscow 119991, Russia

³ Artificial Intelligence Research Institute (AIRI), Moscow, Russia

Abstract. Recently, deep learning has achieved promising results in the calculation of Estimated Time of Arrival (ETA), which is considered as predicting the travel time from the start point to a certain place along a given path. ETA plays an essential role in intelligent taxi services or automotive navigation systems. A common practice is to use embedding vectors to represent the elements of a road network, such as road segments and crossroads. Road elements have their own attributes like length, presence of crosswalks, lanes number, etc. However, many links in the road network are traversed by too few floating cars even in large ride-hailing platforms and affected by the wide range of temporal events. As the primary goal of the research, we explore the generalization ability of different spatial embedding strategies and propose a two-stage approach to deal with such problems.

Keywords: Graph embedding · Machine learning · ETA · Geospatial linked data

1 Introduction

The modern state of traffic induces a remarkable number of forecasting challenges in a variety of related areas. According to the industrial needs, a relevant computation of the estimated time of vehicle arrival can be considered as one of the most actual problems in the logistics domain. In particular, intelligent traffic management systems [15] require significant accuracy in case of arrival time estimation. Besides such an application, computation of ETA also appears as a common issue in the commercial areas which are strongly dependent on optimal routing. The explicit examples of such services are taxi [19], railway [17], vessels [14] and aircraft transportation [3].

Accurate prediction of ETA for cars is a complex task requiring the relevant processing of heterogeneous data. It is frequently represented as time series and graph structure with feature vectors associated with its nodes and/or edges. In comparison with other vehicles, computation of ETA for cars is considerably

V. Porvatov and N. Semenova—Equal contribution.

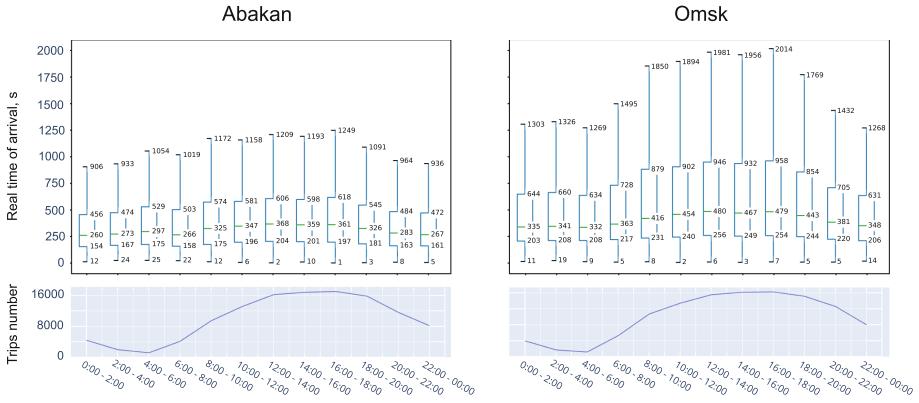


Fig. 1. Demonstration of temporal traffic dynamics: cumulative frequencies of car activity and distribution of trips duration for Abakan and Omsk in the two hours interval.

influenced by the road network topology, nonlinear traffic dynamics, unexpected temporal events, and unstable weather conditions, Fig. 1. The stochastic nature of the introduced problem requires an implementation of a powerful domain-specific regression model with a high generalization ability.

Machine learning proved its outstanding efficiency in a wide range of regression tasks. However, not every model can be efficiently applied to the ETA forecasting due to the mentioned constraints of available data. Previously performed attempts of a simple model implementation (e.g., linear regressions and gradient boosting) were reported as inefficient [11, 27], while the more sophisticated approaches allowed to achieve more optimistic results [21]. Thus, in order to obtain a better performance, we assume the necessity of applying graph neural networks [29] as a part of the presented pipeline.

According to the extensive growth of graph machine learning in recent years, many promising architectures [10, 16] emerged and soon were applied in a wide range of graph-related studies [6, 18]. These models quickly became useful in terms of feature extraction in downstream tasks. Applied to the underlying graph structure of a city road network, such algorithms have the potential to dramatically increase the expressiveness of regression models and therefore should be explored.

In the present paper, we propose and compare different architectures of the hybrid graph neural network for ETA prediction. Our main contributions are the following:

- We introduce and publish the first to our best knowledge dataset¹ with intermediate trip points. This dataset is relevant for consistent ETA prediction task and future usage as a benchmark. We provide common information about trips and city road network as well as road structural properties, marking, and

¹ To receive an access to data you need to send a request to semenova.bnl@gmail.com.

weather conditions (other features are described in Sect. 3 in detail). Additionally, the route data includes auxiliary information which can be used both for evaluation of the ETA and independent prediction of real traveled distance as a separate problem.

- Absence of methodological review of subgraph embeddings in the domain of interest encourages us to overwhelm such a limitation. Instead of focusing on more general approaches which include both spatial and recurrent temporal aspects, we prefer to precisely explore the domain of spatial embeddings as an underdeveloped one at the present moment.
- We conduct a comprehensive evaluation of our method on two real-world datasets which correspond to tangibly different cities. Obtained results of computational experiments motivate us to further develop our research in accordance with achieved significant performance improvements.

2 Related Work

As it has been mentioned above, the ETA-related tasks are a fundamental part of logistic services. In overwhelming number of cases, they demand two properties from the predictive algorithms: computational efficiency and relevant accuracy. The first part of this challenge was unequivocally solved by simple learning models like gradient tree boosting, multi-layer perceptron, and linear regression. However, the quality of these models cannot be reported as sufficient even beyond the commercial logistics.

Along with the simple learning models, deterministic algorithms were also developed in huge amount [2, 26]. In the majority of cases they cannot be compared with learning models in terms of quality. However, some of them were inspiring enough to influence the future development of their concepts in a more sophisticated way.

Limitations of mentioned approaches were partially overwhelmed in DeepTTE [25] and MURAT [13]. The first approach includes a recurrent neural network (RNN) which subsequently predicts the travel time along the trip. As many other recent methods, this algorithm is dependent on intermediate GPS coordinates. At the same time, the second method is closely related to the proposed architecture in the sense of graph embedding usage. In spite of the deep development of the temporal forecasting part, no more than one spatial embedding method was observed in any of this papers.

The most recent studies introduce new solutions with the potential to significantly increase the quality of ETA prediction. WDR [27] is a wide-deep architecture that outperformed a lot of previously established approaches. Its further improvement and computational experiments led the same authors to the design of RNML-ETA architecture [21] which allows to achieve even better results. Simultaneously, another intriguing paper [4] emerged as a prospective modification of ST-GCN methods family [7, 20, 28]. All of these methods use datasets with intermediate points in contrary to the overwhelming majority of early papers. Following this positive trend, we continue studies in the same direction.

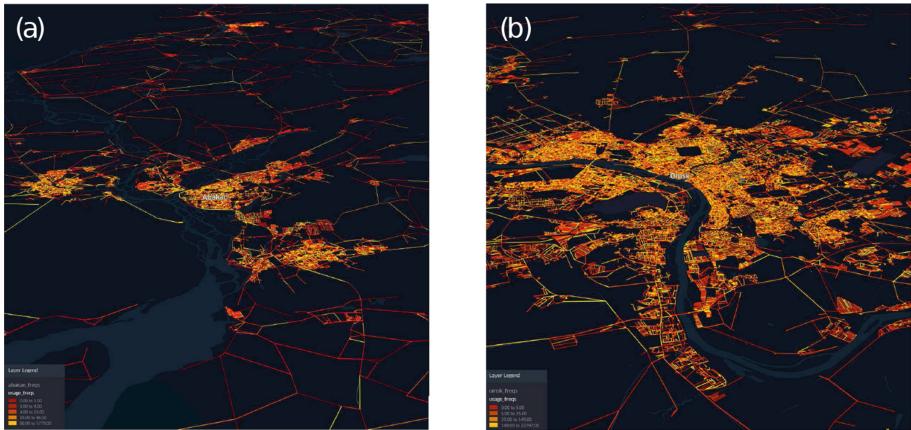


Fig. 2. Edges usage frequencies projected as a heatmap on the road networks of Abakan (a) and Omsk (b). The patterns of edges demand are clearly distinguishable as the topology of networks remains significantly different.

3 Data

In the present work, we use two datasets related to the city networks of Abakan and Omsk. The cities have significantly different scales. Hence, their infrastructure pattern cannot be compared directly. Such a diversity allows us to check the generalization ability of the proposed architectures in a more explicit way. General properties of the dataset are established in Table 1 when the frequencies of road network segments usage are represented in Fig. 2(a,b).

Each dataset consists of both road networks and the routes associated with their edges. City networks contain an abundant number of meaningful features that can be translated to the predictive model in different ways. The route sample includes information about the start and destination point and a set of visited nodes during the ride.

The trip data was collected in the period from December 1, 2020 up to December 31, 2020 by subsidiary companies of Sberbank. A comprehensive description

Table 1. Description of the datasets in terms of common networks characteristics

Property	Abakan	Omsk
Nodes	65524	231688
Edges	340012	1149492
Total trips number	119986	120000
Trips coverage	0.535	0.392
Edges usage median	12	8

of the proposed data is given in Table 2 for the city network and in Table 3 for car routes.

Table 2. Edge features of city network

Feature	Values	Description
Road class	fake road, intra-quarter driveway, dirt road, other city street, main city street, highway, intercity road, federal highway, cycle path, walkway	General road segments categories
Length	\mathbb{Z}_+	Length of a road segment in meters
Width	\mathbb{Z}_+	Width of a road segment in meters
Def speed	{3, 15, 20, 60, 90}	Speed limit on a road section in km/h
Lanes	{0, 1, 2, 3, 4, 5}	Number of lanes in a road segment
Barrier	{0, 1}	Defines the presence of road barriers
Payment flag	{0, 1}	Defines a road segment as toll
Turn restrictions	{0, 1}	Defines an ability to turn on a road section
Pedo offset	{0, 1}	Defines the presence of crosswalk offsets
Bad road	{0, 1}	Defines the condition of a road segment
Style	undefined, archway, crosswalk, stairway, bridge, overground way, invisible, normal, park path, park footpath, subway, pedestrian bridge, underground way, tunnel, living zone, ford	Additional road segments categories

According to the complexity of input data, it cannot be directly translated to a predictive model as an input. In order to correctly solve the desired task, it is recommended to filter the established dataset and perform feature engineering. Trips that have a rebuild count more than 1 should be optionally separated from the main volume of routes as well as anomaly short and long routes. Values of start (finish) point parts and dist to a(b) can be also added or subtracted from the total estimated length of the route in order to obtain a better spatial resolution of subgraph embeddings.

Table 3. Features of trip dataset

Feature	Values	Description
Nodes	$\{\hat{V} \subset V\}$	Subset of nodes
Dist to a	\mathbf{Z}_+	Length of a segment between actual start point and its projection on the first edge
Dist to b	\mathbf{Z}_+	Length of a segment between actual end point and its projection on the last edge
Start point part	\mathbf{Z}_+	Part of the first edge where the trip starts in meters
Finish point part	\mathbf{Z}_+	Part of the last edge where the trip ends in meters
Start UTC	\mathbf{Z}_+	Start time of the trip in UTC format
Real time of arrival	\mathbf{Z}_+	Trip duration in seconds
Real dist*	\mathbf{Z}_+	Actual traveled distance in meters
Rebuild count*	\mathbf{Z}_+	Number of route rebuilds that corresponds to the destination change

4 Methods

The task can be mathematically formulated as a regression problem that extended by a special procedure of an automatic feature engineering. In order to handle this challenge, we generate vector representations of the road segments via GNNs, aggregate them to the trips embeddings and then apply a regression model which predicts ETA.

Given a graph $G = (V, A, X)$ of the city road network, where $V = \{v_1, v_2, \dots, v_n\}$ denotes the set of graph vertexes (road segments), $A: n \times n \rightarrow \{0, 1\}$ denotes the adjacency matrix (each edge encodes connectivity of the road segments), and $X: n \times m \rightarrow \mathbf{R}$ is a matrix of node features.

The goal is to compute such a representation of each node $v_i \in V$ that can be effectively aggregated in accordance with structural properties of the route $s_j := \{v_{j_1}, \dots, v_{j_t}\}$, $s_j \in S$. There are two main aggregation strategies that potentially allow to construct a meaningful route subgraph embedding. The first one based on basic summation of all representations of the nodes that are included to the exact route

$$z_{s_j} = \sum_{i=1}^{\#s_j} Z(v_{j_i}), \quad (1)$$

where $Z(\cdot)$ is the node embedding function.

Another approach related to initial graph extension by virtual nodes. This procedure induces a new graph $\hat{G}(V', A', X')$, where $V' = \{v_1, \dots, v_n, v_{n+1}, \dots, v_{n+\#S}\}$, $A': (n + \#S) \times (n + \#S) \rightarrow \{0, 1\}$, $\forall v_i, v_n \in V$ adjacency matrix defined as $A'(v_i, v_j) = A(v_i, v_j)$. For the other edges, we propose the bijective function $f: V' \setminus V$

$\rightarrow S$ that defines $\forall v'_k \in V' \setminus V$ and $\forall v_l \in f(v'_k)$ values in remaining part of the extended adjacency matrix as $A'(v_l, v'_k) = 1$. In agreement with this method,

$$z_{s_j} = Z(f^{-1}(s_j)). \quad (2)$$

For both strategies it is crucial to find the appropriate node embedding function $Z(\cdot)$ which has a significant impact on the relevance of the final route subgraph representations. We propose graph convolutional networks [10], GAT [23], and GraphSAGE [8] as the main candidates for nodes representation learning. The ideas behind these methods are quite similar as they all encode nodes to vectors of a fixed size via a repeated aggregation over a local neighborhood. However, while the GCN is based on mean aggregation, GraphSAGE pretends to be a more flexible and representative instrument due to its different aggregators and embedding concatenation stage. On the other hand, GAT adopts the mechanism of attention [22] firstly proposed in Natural Language Processing (NLP) to the needs of graph machine learning. To explicitly reveal the relevance of the mentioned approaches, in the following we briefly introduce the main aspects of each method.

Graph Convolutional Network (GCN). For a given graph $G(V, A, X)$ this method defines an effective approach to network information aggregation. Single graph convolution layer is its atomic unit that can be represented as

$$H^{(l+1)} = \sigma \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right), \quad (3)$$

where $l + 1$ is the current convolution layer number, σ is an arbitrary nonlinear function (e.g., ReLU), $H^{(0)} = X$, $\tilde{A} = A + I_N$, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ and W^l is the matrix of learning parameters.

GraphSAGE. This algorithm mostly inherits the notation of convolutions from the GCN architecture, but instead of using full graph it directly computes convolution for each node v in the iterative manner

$$h_v^{l+1} = \sigma \left(W^l \cdot \text{CONCAT} \left(h_v^l, h_{N(v)}^{l+1} \right) \right), \quad (4)$$

where $h_{N(v)}^{l+1}$ can be extracted by a few different aggregate functions for the set of neighbour nodes $N(v)$.

Graph Attention Network. The last considered method is based on the attention mechanism which also avoids transductive GCN constraints and apply the iterative aggregation procedure

$$h_i^{l+1} = \text{CONCAT}_{k=1}^K \sigma \left(\sum_{j \in N(i)} \alpha_{ij}^k W^k h_j^l \right). \quad (5)$$

The attention coefficient is computed as follows:

$$\alpha_{ij} = \frac{\exp \left(\sigma \left(a^T \cdot \text{CONCAT}(W h_i^l, W h_j^l) \right) \right)}{\sum_{k \in N(i)} \exp \left(\sigma \left(a^T \cdot \text{CONCAT}(W h_i^l, W h_k^l) \right) \right)}, \quad (6)$$

where a^T is a transposed vector of attention trainable parameters.

In order to boost the expressiveness of these methods and convert supervised setups to unsupervised, we propose to embed them as a part of the Deep Graph InfoMax pipeline [24]. This approach is based on minimizing of a two-component loss function

$$L = \frac{1}{N+M} \sum_{i=1}^N E_G [\log D(d_i, T)] + \sum_{j=1}^M E_C [\log (1 - D(\tilde{d}_j, T))] \quad (7)$$

which aims to learn how to distinguish initial nodes representations d and corrupted ones \tilde{d} , Fig. 3.

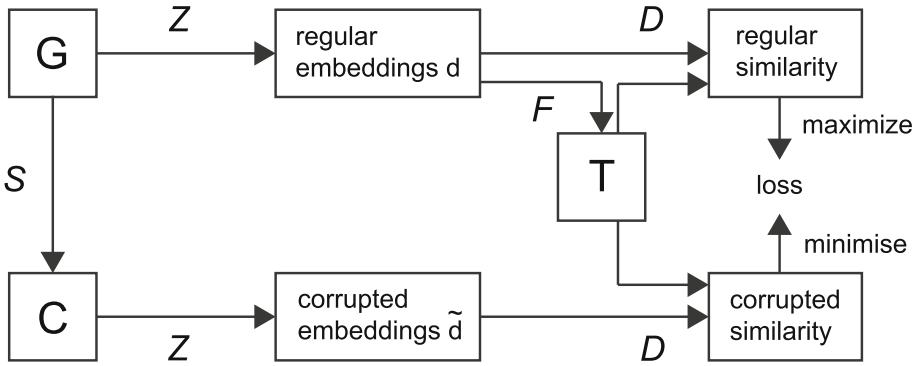


Fig. 3. Deep Graph Infomax corrupts feature vectors of the input graph G by function S (in the used realisation it shuffles features), constructs regular and corrupted node embeddings by applying $Z(\cdot)$, and finally estimates their similarity to the ground-truth vector T by the discriminator function D .

Once embeddings of routes z_{s_j} are computed, each vector can be extended by additional information about the weather conditions and corresponding temporal categorical features. After these manipulations with route vectors z_{s_j} they can be finally fed to the regression model.

5 Results

In order to perform the training and evaluation of proposed architectures, we need to split the datasets into three samples. We trained our model on the first 100 000 trips, while the test and validation steps were performed on equal parts of the remaining datasets.

Following the evaluation standards, we use a common set of metrics for the ETA prediction task: Mean Average Error (Eq. 8), Mean Average Percentage Error (Eq. 9), and Rooted Mean Square Error (Eq. 10).

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - y'_i|, \quad (8)$$

$$\text{MAPE} = \frac{100}{N} \sum_{i=1}^N \left| \frac{y_i - y'_i}{y_i} \right|, \quad (9)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - y'_i)^2}. \quad (10)$$

5.1 Implementation Details

Computational experiments were provided with the use of StellarGraph [5] library. All models were trained on 2 GPU Tesla V100, the total training time of the pipeline for the best models is 9 h. During the embedding construction process, we used three types of each observed architecture with the number of layers from 1 to 3 and the fixed output of size 128. Neural networks weights were trained by Adam optimizer [9] due to its good convergence and stability. We use the static learning rate parameters $L_1 = 0.001$ for node embedding generation and $L_2 = 0.0001$ for regression.

5.2 Experiments

We performed series of computational experiments varying the strategy of subgraph embedding generation and the method of node representation extraction. As the final regression model, we leverage a multi-layer perceptron (MLP). For the purpose of Deep Graph InfoMax tests extension, we also compute the values of the metrics for regular unsupervised GraphSAGE and regression baseline to illustrate the general capabilities of different approaches. The final values of metrics for each configuration are shown in Table 4.

Table 4. Evaluation results on test sample

	Abakan			Omsk		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Baseline(MLP only)	111.05	316.39	27.129	145.819	296.86	25.019
GraphSAGE + VN	111.23	316.82	27.213	146.003	297.028	25.108
GraphSAGE + Sum	96.575	310.114	22.881	129.831	279.773	22.416
DGI(GCN) + Sum	97.927	310.628	23.506	141.017	289.32	24.335
DGI(GAT) + Sum	101.808	313.01	25.737	133.262	283.22	23.175
DGI(GS) + Sum	95.819	309.627	22.622	130.296	280.058	22.593

As it seen from the table, the best performance was achieved by the GraphSAGE setup with Deep Graph InfoMax in the case of Abakan. Meanwhile, common GraphSAGE also demonstrates promising embeddings quality (especially for Omsk) which is slightly different from its DGI modification. The error distributions of the best models for each dataset are shown in Fig. 4.

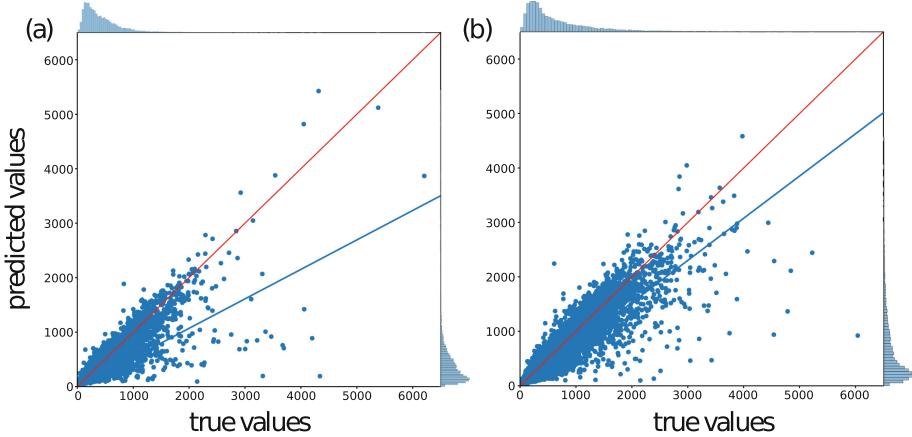


Fig. 4. Error distribution for the regression models trained on Abakan (a) and Omsk (b) datasets.

Unfortunately, the test series of virtual nodes route embeddings turned down our pursuit to report any significant results. We conclude that the expressiveness of this method is limited in the area of interest, despite previous positive attempts of implementation in other tasks [12]. However, such a result was partially foreordained by the studies which also explored subgraph embeddings [1].

6 Conclusion and Outlook

In this work, we implemented and explored a pipeline that includes state-of-the-art algorithms of graph machine learning that emerged in recent years. We trained and tested our model on two consistent datasets which correspond to cities with different road topology types. Our results allow us to conclude that GraphSAGE-based models capture spatial patterns of city networks more substantially.

Our own perspectives include future development and modification of more specific methods based on obtained results. As the primary goal of this research was to find the most efficient methods of subgraph embedding construction in the context of ETA problem, we intend to use this knowledge to construct a more complex spatial approach in the upcoming papers. In the spotlight of our research, we also have an idea to design an powerful generalizing approach to various kinds of road networks with the potential of applying it to a bunch of cities.

Acknowledgements. The work was supported by the Joint Stock Company “Sberbank of Russia”.

References

1. Alsentzer, E., Finlayson, S.G., Li, M.M., Zitnik, M.: Subgraph neural networks. CoRR abs/2006.10538 (2020). <https://arxiv.org/abs/2006.10538>
2. Asghari, M., Emrich, T., Demiryurek, U., Shahabi, C.: Probabilistic estimation of link travel times in dynamic road networks. In: Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2015, Association for Computing Machinery, New York (2015). <https://doi.org/10.1145/2820783.2820836>
3. Ayhan, S., Costas, P., Samet, H.: Predicting estimated time of arrival for commercial flights. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 33–42, July 2018
4. Dai, R., Xu, S., Gu, Q., Ji, C., Liu, K.: Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data. CoRR abs/2006.12715 (2020). <https://arxiv.org/abs/2006.12715>
5. Data61, C.: Stellargraph machine learning library. <https://github.com/stellargraph/stellargraph> (2018)
6. Fout, A., Byrd, J., Shariat, B., Ben-Hur, A.: Protein interface prediction using graph convolutional networks. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS 2017, pp. 6533–6542. Curran Associates Inc., Red Hook (2017)
7. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 922–929, July 2019
8. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 1025–1035 (2018)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2014). [arxiv:1412.6980Comment](https://arxiv.org/abs/1412.6980): Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego (2015)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations, ICLR 2017 (2017)
11. Kviesis, A., Zacepins, A., Komasilovs, V., Munizaga, M.: Bus arrival time prediction with limited data set using regression models, In: Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2018), pp. 643–647, January 2018
12. Li, J., Cai, D., He, X.: Learning graph-level representation for drug discovery (2017)
13. Li, Y., Fu, K., Wang, Z., Shahabi, C., Ye, J., Liu, Y.: Multi-task representation learning for travel time estimation. In: International Conference on Knowledge Discovery and Data Mining (KDD 2018) (2018)
14. Park, K., Sim, S., Bae, H.: Vessel estimated time of arrival prediction system based on a path-finding algorithm. Mar. Trans. Res. **2**, 100012 (2021). <https://www.sciencedirect.com/science/article/pii/S2666822X21000046>

15. Paruchuri, V., Chellappan, S., Lenin, R.B.: Arrival time based traffic signal optimization for intelligent transportation systems. In: 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), pp. 703–709 (2013)
16. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 2014. <https://doi.org/10.1145/2623330.2623732>
17. Prokhorchenko, A., et al.: Forecasting the estimated time of arrival for a cargo dispatch delivered by a freight train along a railway section. East.-Eur. J. Enterp. Technol. **3**, 30–38 (2019)
18. Sanchez-Gonzalez, A., et al.: Graph networks as learnable physics engines for inference and control. CoRR abs/1806.01242 (2018). <http://arxiv.org/abs/1806.01242>
19. Shi, C., Chen, B.Y., Li, Q.: Estimation of travel time distributions in urban road networks using low-frequency floating car data. ISPRS Int. J. Geo.-Inf. **6**, 253 (2017)
20. Song, C., Lin, Y., Guo, S., Wan, H.: Spatial-temporal synchronous graph convolutional networks: a new framework for spatial-temporal network data forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 914–921, April 2020
21. Sun, Y., Fu, K., Wang, Z., Zhang, C., Ye, J.: Road network metric learning for estimated time of arrival (2020)
22. Vaswani, A., et al.: Attention is all you need. CoRR abs/1706.03762 (2017). <http://arxiv.org/abs/1706.03762>
23. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph Attention Networks (2018)
24. Veličković, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D.: Deep graph infomax. In: International Conference on Learning Representations (2018)
25. Wang, D., Zhang, J., Cao, W., Li, J., Zheng, Y.: When will you arrive? Estimating travel time based on deep neural networks. In: AAAI (2018)
26. Wang, Y., Zheng, Y., Xue, Y.: Travel time estimation of a path using sparse trajectories. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014, pp. 25–34. Association for Computing Machinery, New York (2014). <https://doi.org/10.1145/2623330.2623656>
27. Wang, Z., Fu, K., Ye, J.: Learning to estimate the travel time. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2018, pp. 858–866. Association for Computing Machinery, New York (2018). <https://doi.org/10.1145/3219819.3219900>
28. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. CoRR abs/1801.07455 (2018). <http://arxiv.org/abs/1801.07455>
29. Zhou, J., et al.: Graph neural networks: a review of methods and applications. AI Open **1**, 57–81 (2020)



What Is Learned in Knowledge Graph Embeddings?

Michael R. Douglas^{1,2(✉)}, Michael Simkin¹, Omri Ben-Eliezer^{1,3}, Tianqi Wu^{1,4}, Peter Chin^{1,5}, Trung V. Dang⁵, and Andrew Wood⁵

¹ CMSA, Harvard University, Cambridge, MA, USA
`{mdouglas,msimkin}@cmsa.fas.harvard.edu`

² Department of Physics, YITP and SCGP, Stony Brook University,
Stony Brook, NY, USA

³ Department of Mathematics, MIT, Cambridge, MA, USA
`omrib@mit.edu`

⁴ Departments of Mathematics and Computer Science, Clark University,
950 Main Street, Worcester, MA, USA
`tianwu@clarku.edu`

⁵ Department of Computer Science, Boston University, Boston, MA, USA
`{spchin,trungvd,aewood}@bu.edu`

Abstract. A knowledge graph (KG) is a data structure which represents entities and relations as the vertices and edges of a directed graph with edge types. KGs are an important primitive in modern machine learning and artificial intelligence. Embedding-based models, such as the seminal TRANSE [Bordes et al. 2013] and the recent PAIRRE [Chao et al. 202] are among the most popular and successful approaches for representing KGs and inferring missing edges (link completion). Their relative success is often credited in the literature to their ability to learn logical rules between the relations.

In this work, we investigate whether learning rules between relations is indeed what drives the performance of embedding-based methods. We define motif learning and two alternative mechanisms, network learning (based only on the connectivity of the KG, ignoring the relation types), and unstructured statistical learning (ignoring the connectivity of the graph). Using experiments on synthetic KGs, we show that KG models can learn motifs and how this ability is degraded by non-motif (noise) edges. We propose tests to distinguish the contributions of the three mechanisms to performance, and apply them to popular KG benchmarks. We also discuss an issue with the standard performance testing protocol and suggest an improvement.

1 Introduction

1.1 Definitions and Basic Properties of KGs

A knowledge graph (henceforth abbreviated KG) is a graph-structured data model, often used to store descriptions of entities such as people, places, and

events. A KG can be defined as a collection of ordered triples $(h, r, t) \in V \times T \times V$, where V is the set of entities and T is the set of relation types. As an example, consider the relation “John lives in Chicago”; here $h = \text{“John”}$, $r = \text{“lives in”}$, and $t = \text{“Chicago”}$.

In graph theoretic terms, a KG is a directed graph in which each edge has both an orientation and a type (or label). Its vertices correspond to entities, and each triple (h, r, t) corresponds to an edge. We will also use the notations $h \xrightarrow{r} t$, $t \xleftarrow{r} h$ and $(t, -r, h)$ to denote the triple (h, r, t) , and $h \not\xrightarrow{r} t$ to denote absence of a triple.

KGs are more flexible than traditional database models, but more structured than text, which facilitates automated reasoning. Many represent “general knowledge,” including Wikidata, YAGO [18] and DBpedia [17]. There are many more KGs covering specialized topics, such as SemMedDB for biomedical data [16]. KGs have also been created as benchmarks to test KG software. The KGs in our experiments came from the Open Graph Benchmark (OGB) collection [13] and the OpenKE compilation [11]. The largest of these is the OGBL-WIKIKG2 benchmark. It is based on Wikidata, and has 17,137,181 edges of 535 types connecting 2,500,604 entities.

1.2 Embedding Models

Much of the success in learning KGs is due to *embedding models*. Such models identify the vertices with points in a metric space with and interpret proximity under a relation-type-dependent transformation as graph adjacency (with the corresponding edge label). The literature describes many embedding models for knowledge graphs, including: TRANSE [8], ROTATE [21], PAIRRE [10]. For simplicity and concreteness in this work we primarily consider TRANSE and PAIRRE, due to their state-of-the-art performance. While the models are similar that similar reasoning can be applied to their analysis, they differ in expressiveness, as highlighted in [10]. For results on the expressive power of TRANSE, we refer the reader to [6].

In what follows, G denotes a KG with vertex set V , edge types T , and relations R . Both TRANSE and PAIRRE consist of an underlying vertex map $\psi : V \rightarrow \mathbb{S}^{k-1}$.¹ They differ in their approach to modelling relations.

In TRANSE for each $r \in T$ the model learns a vector $\mu_r \in \mathbb{R}^k$. The idea is that for every triple (h, r, t) , it holds that $\psi(h) + \mu(r) \approx \psi(t)$ if and only if (h, r, t) is a relation in G .

In PAIRRE for each $r \in T$ the model learns two vectors $\mu_r^h, \mu_r^t \in \mathbb{R}^k$. The idea is that $\mu_h(r) \circ \psi(h) \approx \mu_t(r) \circ \psi(t)$ if and only if (h, r, t) is a relation in G . Here \circ denotes the Hadamard (elementwise) product $(a \circ b)_i = a_i b_i$.

KG embedding models are trained using standard ML techniques, by gradient descent on the error with which these defining properties hold. One slightly nonstandard point is that negative sampling is used to estimate the error for the triples not in G (negative edges or simply negatives). Recent works use self-adversarial sampling [21] in which the negatives are sampled by making use of the model being trained.

¹ Many implementations of TRANSE embed into \mathbb{R}^k . Also, the norms below vary but are often ℓ_1 .

As detailed in the next section, there are two common applications for KGs: Binary link classification and ranking of potential link completions. For the classification task, the models learn a threshold $\gamma > 0$. A triple (h, r, t) is predicted to be in G if and only if $\|\psi(h) + \mu_r - \psi(t)\| < \gamma$ (in the case of TRANSE) or $\|\mu_r^h \circ \psi(h) - \mu_r^t \circ \psi(t)\| < \gamma$ (in the case of PAIRRE). In the ranking task the model is given a head $h \in V$, a relation $r \in T$, and a list of potential completions $t_1, \dots, t_n \in V$. The model ranks the likelihood of the relations $\{(h, r, t_i)\}_{i=1, \dots, n}$ in increasing order of $\|\psi(h) + \mu_r - \psi(t_i)\|$ or $\|\mu_r^h \circ \psi(h) - \mu_r^t \circ \psi(t_i)\|$ for TRANSE and PAIRRE, respectively.

1.3 Tasks and Evaluation

The most studied KG task is link completion: given partial information about a relation, say its type and the head, find the “valid” tails (or given the tail and type, find the heads). We stress that in contrast to link prediction in network theory, the type of the relation is important.

What does “valid” mean? If we take it to mean edges which are already in the graph, then this is a problem of database retrieval. One could instead consider link completion as a particular type of knowledge graph completion. This refers to tasks such as filling in missing entries in a KG, correcting errors, predicting the time evolution of a KG, or other forms of inference, all with the common assumption that there is an implicit “ground truth” KG to which the dataset is an approximation. A simple example would be a dataset which is a sample from a known KG. In this case, the valid completions would be those which appear in the complete KG. The hypothesis is that the KG has structure which can be learned and used to make these predictions.

Here are some examples of structures in Wikidata which could be used to make link predictions:²

- Property P47, “shares border with,” relates pairs of geographic regions. It is symmetric, so from $A \xrightarrow{P47} B$, we can deduce $B \xrightarrow{P47} A$.
- Property P103, “native language,” implies property P1412, “language(s) that a person speaks, writes or signs, including the native language(s).”
- Property P131, “located in the administrative territorial entity,” is transitive.

We will refer to structures of this type, which impose relations between the relations, as “rules,” and discuss them more systematically below.

As a more complicated example, by combining P937, “work location,” which relates people and places, with P37, “official language,” which relates places and languages, we could hope to deduce P1412, “languages spoken, written or signed.” This would be a statistical rather than a logical inference, but a very likely one. A further complication is that P937 links are supposed (by the Wikidata guidelines) to be as specific as possible, so we might need to use P131 to make the inference as well. One can see that there is a large scope for this type

² Lists and descriptions of the Wikidata entities and relations are readily available, try for example <https://www.wikidata.org/wiki/Q5>.

of inference, and that the number of rules, each of which would need to be programmed in a traditional approach, is also large. The prospect of automatically learning these rules is very attractive.

To evaluate these ideas, one needs a test dataset of triples and a measure of link completion accuracy. One can of course split a larger dataset into training and testing sets by sampling. While this is very standard in ML, for a prediction problem it can be criticized on the grounds that the model can take advantage of structure only visible once the completions are known. To avoid this criticism, the `ogbl-wikikg` benchmark did its split by sampling the Wikidata information at three different dates, and then using the links added during the two intervals as the validation and test datasets. This has the potential problem that the data addition process might be nonstationary (time dependent).

The most popular measure of accuracy is defined as follows (for tail prediction; head prediction is analogous). For each testing triple (h, r, t) , we give h and r to the model, which gives us a list of candidates for t ranked by score. The `Hits@N` metric is then the fraction of triples for which the correct t has rank N or higher, and the `MRR` (mean reciprocal rank) metric is the mean of $1/\text{rank}$ over the test set.

For large KGs the full list of candidates for t is expensive to evaluate, so in practice one often considers a subset. This is usually chosen by filtered uniform sampling, meaning that candidate completions t' are uniformly sampled from the vertex set excluding those for which the testing or training dataset contains the triple (h, r, t') . The OGB benchmark evaluates both head and tail completion, each with 500 filtered negatives, and reports the results for the combined test set.

Looking at the OGB leaderboard³ and Table 3, the OGBL-WIKIKG2 dataset can be completed using TRANSE (with 500 dimensional embeddings) to get testing MRR 0.43 and `Hits@1` 0.41. In other words, without knowing anything about the relations other than the graph, given an entity and relation type, this simple model can predict in over 40% of cases the other entity involved in the relation. Taken at face value, this is remarkable. How does this work? What structure in the dataset is being used?

1.4 Do KG Models Learn Rules?

A popular hypothesis is that the KG models are learning rules which are usually satisfied by the relations. The simplest rules involve pairs of links: symmetry ($A \xrightarrow{1} B \Leftrightarrow B \xrightarrow{1} A$), exclusivity ($A \xrightarrow{1} B \Rightarrow A \xrightarrow{2} B$), and subrelation ($A \xrightarrow{1} B \Rightarrow A \xrightarrow{2} B$). Other rules involve multiple links, such as the conjunction rule,

$$A \xrightarrow{1} B \xrightarrow{2} C \Rightarrow A \xrightarrow{3} C. \quad (1)$$

Many KG works advocate a model by showing its ability to express these rules. For example, in TRANSE the rule Eq. 1 is naturally expressed by the embedding property

³ ogb.stanford.edu/docs/leader_linkprop.

$$\mu_1 + \mu_2 \approx \mu_3, \quad (2)$$

as then $\psi(B) \approx \psi(A) + \mu_1$ and $\psi(C) \approx \psi(B) + \mu_2$ will imply $\psi(C) \approx \psi(A) + \mu_3$.

Now it is not *a priori* obvious that KG completion is operating by learning and using these rules. There might be other structures in the dataset, such as the clustering which is much studied in graph theory, which are responsible. It might also be that while rules can be learned in principle, the real world KGs do not have high enough signal to noise to do this.

Let us preview some experiments which bear on these questions (see Sect. 4 for details):

- Use TRANSE for the KG completion task, but learn only the vertex embeddings and freeze the relation embeddings. This gets almost the same MRR as the original model.
- Replace all the edge labels (in both training and testing datasets) with a single label. Now the MRR drops, but only from 0.43 to 0.36.

Since we expect these modifications to drastically handicap rule learning, such results cast doubt on the idea that rule learning by learning properties such as Eq. 2 is the main explanation of KG model performance.

1.5 An Issue with Evaluating Large KGs

Before we take these unexpected results too seriously, we should ask to what extent they might be explained by problems with the data or evaluation procedures.

The practice of using a sampled list of negatives is a shortcut which does not correspond to a real KG task, so it should be justified by comparison with the “true metrics” computed using a complete list. As we will see in Sect. 4, while 500 negatives is adequate for our other KGs, it is quite small for OGBL-WIKIKG2. We noticed this by evaluating a baseline (or “null”) model which (for tail completion) ignores the head and takes the score of (h, r, t) to be the conditional probability $P(t|r)$ estimated on the training data as

$$P(t|r) \equiv P(\text{tail}|\text{rel}) \sim \frac{\text{Number of edges } x \xrightarrow[r]{} t \forall x}{\text{Number of edges } x \xrightarrow[r]{} y \forall x, y}. \quad (3)$$

This simple model gets an MRR of 0.23 for tail prediction.

To understand why, consider the relation P1412, “languages spoken, written or signed.” There are about 250 entities in Wikidata which represent languages, of which only a few are common. But since there are about $2.5 \cdot 10^6$ vertices, the probability that a uniform sample of size 500 will contain even a single language entity is $250 \cdot 500 / 2.5 \cdot 10^6 \sim 0.05$. So, with very high probability, the correct result will rank first, just because it is the only language entity on the list.⁴

⁴ It also turns out that P1412 is over-represented in the test set. In all, it contributes about .10 of the total tail MRR = 0.23 of the simple model.

Looking at all the items in the OGBL-WIKIKG2 test set, only about 10% of the entries have any negative vertices with the correct relation type. This suggests that the true metrics could be rather different. We could still use these uniformly sampled metrics to compare models, if they are monotonic in the true metrics. Even so, one might lose discriminatory power.

These general observations are not new to us; in Sect. 1.7 we cite several works which point out the need for the testing procedure to use plausible negatives and propose ways to get them, using either human input or another inference procedure to create the negatives. A direct but costly way to solve the problem is to increase N_{neg} , in this example by a factor somewhat larger than 20.

A simpler way to mitigate the problem, new so far as we know, is to sample the negatives using the model Eq. 3. In Sect. 4 we use a 50-50 mixture of this sampling with uniformly sampled negatives, and compare these resampled results with uniform sampling. As an example, the OGBL-WIKIKG2 resampled (or R-) MRRs for TRANSE and PAIRRE are 0.13 and 0.28 respectively.

On re-evaluating the results from Sect. 1.4, we find that freezing relation embeddings reduces the R-MRR from 0.13 to 0.09, and removing edge labels reduces it to 0.06. So this is part but not all of the resolution.

1.6 Summary of Our Contributions

Our main contribution is to propose a way to study the question “What do KG models learn?”. We define three types of learning and propose tests to distinguish their contributions to performance. Motif learning is a precise definition of the rule learning posited in many KG works, which depends only on structures in the KG. Network learning is based purely on connectivity, and unstructured statistical learning uses a graphical model which ignores network structure.

We find this distinction useful for several reasons. First, it clarifies the interpretation of experiments. The standard benchmark KGs have different statistical properties and this is reflected in different potential performance for the three learning mechanisms. Rather than say that one KG is better than another (after all the goal is to work with general KGs), we can factor out these differences and make a combined interpretation of results. As for the unexpected results, their interpretation is clearer once one realizes that the link completion task can be solved in different ways. Second, network learning and graphical models are classic topics and are far better understood than the general problem of KG learning. By seeing how they fit into the general problem, we make a principled start on bringing the general theory up to the same level. For example, we can use the mathematics of graph embeddings to understand network learning.

By study of synthetic KGs, we show that popular KG embedding models can do motif learning, and study how this degrades with noise and other features of the problem. The mathematics of graph theory predicts a phase transition at a critical noise threshold, and we exhibit this. We can also study freezing relation learning in a controlled setting and argue that the remaining performance is due to network learning.

1.7 Related Work

Given the relative success of embedding-based methods for knowledge representation, there have been many works explaining the efficacy of these methods from various perspectives; here we describe some representative and closely related works. [19] and [22] point out that current metrics for evaluating KG methods have significant flaws and suggest alternative evaluation procedures. [3] aim to understand the latent structure of knowledge graph embeddings by leveraging insights from word embeddings. They import semantic concepts from the natural language processing literature, such as paraphrases, analogies, and context shifts, and find evidence that these concepts also play a role in some relation types of KG embeddings. [15] and [14] demonstrate that simple baselines can sometimes outperform much more complicated KG embedding techniques, which is in line with many results in this paper. [2] suggest that many of the most popular benchmarks used to evaluate embedding methods contain significant redundancies and are thus not sufficiently challenging to capture the difficulties arising with real-world data. [9] initiate an investigation of the geometry of different types of embedding methods.

2 Three Mechanisms of KG Learning

Besides learning rules, what other structure could the KG models be using? Let us state two alternate hypotheses, and then restate rule learning as motif learning, a definition which only uses KG structure. For all three, their precise definition will be in terms of a restriction on the information which can be used in the mechanism.

2.1 Unstructured Statistical Learning

A simple first hypothesis is that the models are not learning the network structure, rather they are picking up on statistical information such as Eq. 3. There are many more sophisticated models of this type, such as [23]. A broad class are covered by

Definition 1. *Unstructured statistical learning models the probability distribution of triples (h, r, t) in terms of latent variables l_h, l_r, l_t as*

$$P(h, r, t) = \sum_{\substack{l_h \in \mathcal{L}_h \\ l_r \in \mathcal{L}_r \\ l_t \in \mathcal{L}_t}} P(h|l_h)P(r|l_r)P(t|l_t)P(l_h, l_r, l_t). \quad (4)$$

This is a standard graphical model [7] which can easily learn constraints of this type, but cannot learn rules or network structure.⁵ There are variants which

⁵ The perceptive reader will note that as stated this is false, with the simplest counterexample being to identify $\mathcal{L}_h \cong \mathcal{H}$ and take $P(h|l_h) = I[h = l_h]$, etc. It is surprisingly difficult to make this constraint precise, and we plan to do this elsewhere. For present purposes we approximate it by requiring $|\mathcal{L}_h| \ll |\mathcal{H}|$, etc.

can learn symmetry, subrelation and exclusivity in terms of joint probabilities of relations with the same head and tail.

How far can this idea go towards explaining link prediction results? We will discuss the general model of this type elsewhere. If we take the latent variables to be class probabilities, these look rather similar to embedding models such as TUCKER [5].

Here we consider an embedding version of the “null model” Eq. 3, which we call RE.⁶ It has separate embeddings ψ_h for heads, ψ_t for tails and μ_r for relations. The score of a tail completion (h, r, x) is simply $\psi_t(x) \cdot \mu_r$ (or a normalized version of this). Good performance of this model may tell us more about a dataset than about KG learning, but this illustrates the idea.

2.2 Network Learning

Our next hypothesis is that the models are using the network structure, but only its connectivity, ignoring the relation types.⁷ This certainly seems to fit with the results in Sect. 1.4!

For example, if there are many candidates for a tail vertex, a network model might prefer the ones closest to the head. Arguably the simplest definition of “closest” is the vertices which minimize the number of edges in the shortest path, independent of orientations. One could propose other definitions, assigning lengths to edges which might depend on node degrees and/or orientation. This type of proximity structure is easily captured by an embedding model, indeed the topic of metric and similarity embeddings of graphs is very well developed, with many reviews including [12]. We certainly expect that the KG embedding models use proximity as a factor, but to what extent does proximity explain their performance?

As a straw man hypothesis, suppose that we make the predictions by uniformly sampling the distance two neighborhood. This does very poorly for two reasons. First, the degree two neighborhood of a KG regarded as an undirected graph tends to be very large, because of the presence of tail nodes of very high degree.⁸ This might be dealt with by redefining proximity to exclude such nodes, but even a neighborhood of size $(\text{average degree})^2$ is too large for this to work by itself. However, the combination of proximity with unstructured statistical learning might not be a bad model. Can we define its separate contribution?

To make this precise, we make the following definition.

Definition 2. *Network learning can use any directed graph structure which does not depend on edge types.*

⁶ Following the convention in which KG embedding models have names ending with the letters capital R and/or E.

⁷ We use the term “network” rather than “graph” at this point to reduce confusion with statistical terminology, in particular “graphical models.”.

⁸ While the average degree of a node in `ogbl-wikikg` is 12, the highest degree node connects to almost 9% of the other nodes. It is Q5, “human,” due to relations such as “Albert Einstein is a human.”.

This includes degree distributions, distance distributions, spectral properties and even frequencies of motifs defined without regard to edge types. A variant would further restrict to undirected graph structure.

We can then compare models allowed to use both network and unstructured information, with those using either separately.

2.3 Inference of Rules by Learning Motifs

From a graph theoretic point of view many rules (though not all, for example disjunction) are related to motifs, small labeled digraphs which appear as subgraphs of the KG. For example, the symmetry rule is related to the motif consisting of both orientations of an edge. The conjunction rule Eq. 1 is related to a triangle motif, a graph with the three vertices A, B, C and the three directed edges corresponding to the three relations. Denote the triangle graph with these edge types as

$$T(1, 2, -3) \equiv \{A \xrightarrow{1} B, B \xrightarrow{2} C, C \xleftarrow{-3} A\}. \quad (5)$$

Note that this motif does not carry exactly the information of the rule. Whereas Eq. 1 treats $A \xrightarrow{3} C$ specially, one could distinguish one of the other edges to get similar but different rules. However, if we grant that the various rules related to the motif have similar probabilities, then the problem of learning motifs will be a good approximation to that of learning rules. There are many works on identifying and learning motifs statistically, with a much studied example being the planted clique problem [4].

To make these ideas precise, we make

Definition 3. *A k -motif model can base its predictions on the statistics of labeled directed subgraphs of the KG with up to k vertices and on the corresponding neighborhood of the given vertex.*

As an example, a 3-motif model which learns Eq. 5 could identify a copy of the LHS of Eq. 1 with $h = A$ and $r = 1$, and predict the RHS.

2.4 Distinguishing the Three Types of Learning

We just outlined three types of learning – of unstructured statistics, of network structure, and of motif structure. There might be other learning mechanisms as well, and architectures suited to them. A clear case is disjunction, which is not a motif.⁹ KG models which work with disjunction often introduce other structures such as “boxes” in embedding space [1, 20].

The mechanisms are not exclusive, indeed one could argue that KG embedding models provide elegant combinations of all three. Still, to properly interpret results and judge models, it is useful to distinguish between them. For example, attributing performance differences to rule learning may be misleading if the other mechanisms have comparable or larger effects.

⁹ Some cases of disjunction can be represented by sets of motifs.

We defined the learning types in terms of conditions on the information they can use. Now TRANSE, PAIRRE and the other embedding models do not satisfy any of the three conditions, so all three mechanisms might be important. Thus, we now ask: how can we distinguish the contributions of these different mechanisms to the performance of a model?

First, the definitions suggest ablation tests:

- By removing labels, we only allow network learning.
- By freezing the relation embeddings, we disable most of the proposed mechanisms for motif learning.
- By adding random “noise” edges with all new relation types, unstructured and motif learning should be hardly affected, while network learning should be degraded.
- Suppose we remove every aspect of the model which relates heads and tails, say by using separate embedding functions for heads and tails. This should degrade network and motif learning much more than unstructured learning.

In all of these cases, the resulting degradation could be interpreted as a measure of the contribution of the affected mechanisms. One has to be careful as ablations such as freezing weights could cause more general degradation, say if the initialization values are inappropriate.

Another class of tests is to look at expected properties of the embeddings. Claims such as “In TRANSE, the relation Eq. 1 is learned by finding relation vectors which satisfy $\mu_1 + \mu_2 \sim \mu_3$,” can be checked directly. As another example, in [6] it is shown how to construct TRANSE embeddings (without relation types) by starting with a metric embedding of the undirected graph obtained by forgetting orientations, and adjoining a dimension. To the extent that this picture is realized, it clearly shows that TRANSE is learning the network structure.

3 Synthetic KGs

One approach to studying the mechanisms involved in KG learning is to analyze the performance of various models on *synthetic* KGs. We present several case studies in which we analyze both performance and specific parameters of the embeddings obtained by analyzing synthetic KGs.

Each synthetic KG we consider has the form $G \cup R$, where G is a (highly structured) deterministic KG and R is a random KG which we think of as noise. In all our examples we take $R = \cup_{r \in T} R_r$, where T is the set of relation types and each R_r is an independent random directed graph on $V(G)$ with each edge present independently with probability $p_r \in [0, 1]$, and all edges have relation type r .

For each graph $G \cup R$ we study the link completion task as discussed earlier with $N_c = 500$ for both head and tail prediction. For the training set we take R as well as a random 0.8-fraction of G . For the testing set we take the remaining 0.2-fraction of G . We exclude R from the test set because for the distributions we use, test edges of R would be information-theoretically unlearnable.

We did a suite of link completion runs with various models, scanning the embedding dimension and the hyperparameter γ . Results quoted are the best train and test MRR and minimal embedding dimension required for this test MRR result.

3.1 Link Completion Without Motif Learning

Table 1. MRR and motif norm for graphs G_1, G_2, G_3, G_4 with models TransE and TransE*. The motif norm of $T(i, j, -k)$ is the ratio of $|\mu_i + \mu_j - \mu_k|$ to the average norm of a translation vector. All norms are ℓ_2 . G_4 contains two motifs, and both norms are reported.

Graph Model	G_1 TransE	G_1 TransE*	G_2 TransE	G_2 TransE*	G_3 TransE	G_3 TransE*	G_4 TransE	G_4 TransE*
MRR	0.62	0.58	0.59	0.50	0.48	0.25	0.16	0.05
Motif norm	0.022	1.45	0.063	1.52	0.727	1.78	0.947	1.65

In our first example we consider graphs with $n = 10000$ vertices, defined as follows: Let G_1 be a graph with 2499 disjoint copies of the triangle motif $T(1, 2, -3)$ and the remaining 2503 vertices isolated. For $i = 1, 2, 3$ and $p \in [0, 1]$, let $R_i(p)$ denote the random graph where each (directed) edge is present independently with probability p , and all have label i . Define:

$$\begin{aligned} G_2 &= G_1 \cup R_1(1/(4n)) \cup R_2(1/(4n)) \cup R_3(1/(4n)), \\ G_3 &= G_1 \cup R_1(3/(4n)) \cup R_2(3/(4n)) \cup R_3(3/(4n)). \end{aligned}$$

We also define the graph G_4 by letting G be the graph with 999 and 499 respective copies of the triangle motifs $T(1, 2, -3)$ and $T(4, 5, -6)$ (all disjoint), and setting

$$G_4 = G \bigcup_{i=1,2,3,7,9} R_i(8/(5n)) \bigcup_{i=4,5,6,8,10} R_i(4/(5n)).$$

To test the extent motif-learning plays a role in modelling these graphs, we trained two embedding models: TRANSE and a variant, TRANSE*, in which the relation vectors are frozen at their random initialization (so that the optimization is only over the vertex embedding). We report the results in Table 1, which we now discuss.

We see that for TRANSE, the model succeeds in the edge-prediction task quite well (although performance clearly degrades with noise). We remark that for these examples, an MRR of ≈ 0.64 is the best one could hope for. This is because for each test edge we only expect the algorithm to make a correct prediction if the other two edges in its motif are included in the training set. The latter event has probability $0.8^2 = 0.64$. Another striking feature is that

MRR for G_4 is approximately one-quarter of the theoretical maximum, despite it having approximately 20 times more noise edges than motif edges. Finally, for each of these graphs, the motifs themselves are learned quite well. This is evidenced by the small norm (relative to the average norm of translation vectors) of $\mu_i + \mu_j - \mu_k$ for each motif $T(i, j, -k)$.

We cannot expect TRANSE* to learn the motifs themselves. Nevertheless, the MRR for the link-prediction task remains quite high. This is perhaps due to alternative learning mechanisms, such as network learning.

Table 2. Statistics of knowledge graphs. nedges denotes the training set, and ntri denotes the number of triangle motifs Eq. 5, for the training set and for training+test. The RE column is the MRR for that model, and the Δ columns are explained in the text.

Dataset	nentity	nrels	nedges	ntri(train)	ntri(all)	RE U/R-MRR	Δm	ΔR
wn11	38193	10	112581	10033	13420	12.7/2.8	0.30	6.2/4.2
wn18	40942	17	141442	25144	27111	10.4/2.7	0.14	7.4/16.4
fb15k	14950	1344	483142	4865600	6100776	25.3/14.2	25.57	22.8/51.4
fb15k237	14504	236	272115	1867991	1898714	29.8/17.9	1.13	18.9/25.0
ogbl_wikikg2	$2.5 \cdot 10^6$	534	$1.6 \cdot 10^7$	10471397	11481634	42.3/13.3	0.63	7.2/6.3

3.2 Motif Factor with Noise

For our second example we explore the addition of noise to a graph with a strongly represented motif by studying the performance of TRANSE on $G_1 \cup R_4(c/n)$, for values of c in the interval $[0, 6]$. We depict our results in Fig. 1.

We first note that for all the graphs, $|\mu_1 + \mu_2 - \mu_3|$ is small. In other words, the motif $T(1, 2, -3)$ is learned. Thus, incorrect predictions are due to the vertex embedding. For $c \leq 0.5$, `hits@1` is near its maximal value of 0.64. Above this value, this measure begins degrading. Curiously this phase transition seems to occur together with the appearance of a linear-sized strongly connected component in the underlying undirected graph (see Fig. 1). We wonder if this component is difficult to embed, leading to deteriorating performance. Alternatively, perhaps there are *local* explanations for incorrect predictions: As c increases, there are more non-motif triangles (i.e., those including edge label 4). If proximity and local structure play a substantial role in making predictions, non-motif triangles could cause incorrect predictions.

4 Experiments on Real KGs

We used 5 datasets in our experiments, OGGL-WIKIKG2 and 4 KGs from OpenKE [11]. Five modified versions of the KGs were considered, one without relation types (XR), and four with noise. Much as in Sect. 3, the noise consists of randomly added edges whose number is a specified multiple from 0.2 to 2.0 of the number of training edges. Table 2 contains their general statistics.

Table 3. Uniform/resampled MRR by dataset and model (in percent). The columns are the baseline, varying numbers of noise edges (expressed as a fraction of the number of training set edges), and XR is the graph without relation types.

		Base	0.2	0.5	1.0	2.0	XR
fb15k	ComplEx	70.3/82.1	67.7/60.1	61.0/47.0	54.0/38.0	41.9/27.1	27.0/12.0
	DistMult	71.0/75.9	63.0/48.4	51.0/33.0	51.0/33.0	41.6/24.5	53.0/51.0
	PairRE	68.0/81.0	68.3/78.0	68.0/72.0	67.0/62.0	63.7/52.5	50.4/40.1
	RotateE	68.2/83.0	68.2/75.7	66.0/64.0	66.0/64.0	56.5/52.8	50.0/38.0
	TransE	68.0/81.0	69.2/80.7	70.0/78.0	70.0/74.0	70.0/74.2	46.0/30.0
fb15k237	ComplEx	51.0/42.0	48.1/33.0	44.0/28.0	36.0/22.0	23.3/14.2	31.0/12.0
	DistMult	51.0/41.0	47.0/29.5	42.0/25.0	38.0/21.0	29.1/16.0	26.0/13.0
	PairRE	49.0/38.0	47.9/35.7	47.0/35.0	47.0/34.0	46.1/29.7	30.0/12.0
	RotateE	50.0/39.0	43.8/32.2	40.0/28.0	33.0/23.0	21.3/16.3	32.0/13.0
	TransE	51.0/39.0	50.3/38.3	51.0/38.0	50.0/38.0	50.4/37.7	32.0/14.0
wn11	ComplEx	28.0/28.0	20.8/20.7	15.0/15.0	9.0/ 8.0	5.6/ 3.7	20.0/20.0
	DistMult	27.0/27.0	18.1/17.5	11.0/11.0	7.0/ 5.0	3.7/ 2.6	20.0/20.0
	PairRE	21.0/15.0	17.4/11.8	17.0/10.0	20.0/13.0	15.2/ 6.7	25.0/19.0
	RotateE	28.0/20.0	18.3/11.6	12.0/ 8.0	8.0/ 5.0	4.1/ 3.1	21.0/17.0
	TransE	33.0/22.0	24.5/11.9	21.0/ 8.0	17.0/ 6.0	14.5/ 4.3	27.0/18.0
wn18	ComplEx	92.0/96.0	90.7/95.8	90.0/95.0	89.0/93.0	81.2/81.1	89.0/93.0
	DistMult	91.0/96.0	90.3/94.6	89.0/93.0	86.0/85.0	60.9/54.5	90.0/93.0
	PairRE	81.0/80.0	89.2/89.2	90.0/94.0	90.0/94.0	79.7/69.4	89.0/88.0
	RotateE	72.0/66.0	90.1/93.9	90.0/94.0	89.0/93.0	88.4/90.8	87.0/85.0
	TransE	88.0/88.0	89.5/91.6	89.0/91.0	87.0/87.0	76.9/64.1	80.0/72.0

Table 4. MRRs for OGBL-WIKIKG2 by model

	Base	XR
ComplEx	48/17.0	37/11.6
DistMult	44/11.0	45/14.3
PairRE	56/28.0	39/10.2
RotateE	43/18.0	40/ 9.8
TransE	44/13.0	37/ 6.4

In Tables 3 and 4 we give performance results for 5 models (TRANSE, PAIRRE, DISTMULT [25], COMPLEX [24] and ROTATE [21]), evaluated using the OGB code. We did some optimization of the hyperparameter gamma, while the hidden dimension was taken large (400 and 1000 for the smaller KGs and 500 for OGBL-WIKIKG2). Standard and resampled MRRs are given together as U-MRR/R-MRR. Comparing the two, the similarity of their rank-orderings can be measured by the Spearman rank correlation coefficient. This is around 0.9 for the smaller KGs but drops to around 0.6 for OGBL-WIKIKG2.

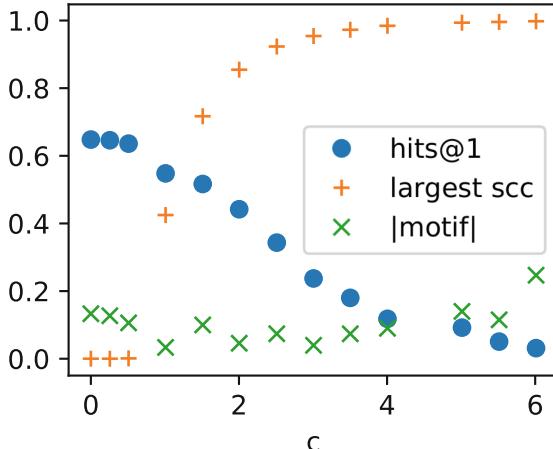


Fig. 1. hits@1, largest strongly connected component, and ratio of $|\mu_1 + \mu_2 - \mu_3|$ to average translation vector norm (in ℓ_1), as functions of c .

4.1 Interpretation

Do these results fit with the hypothesized three mechanisms? Let us point out some suggestive patterns.

First, consider the propensity of each KG to each type of learning. KGs with more relations favor unstructured learning, and this is apparent in the RE model performance. KGs with more motifs favor motif learning. In Table 2, the column Δm is the number of triangle motifs which contain test set edges over the number of test set edges. while ΔR is the difference between TransE base and XR R-MRR. Presumably, dependence on network learning will show up in dependence on noise. This dependence is strongest for WN11, which has the fewest relations.

Next, looking at the dependence on the model, COMPLEX and DISTMULT work best for the KGs with few relations (WN11 and WN18), and are significantly more affected by noise than the others. This is consistent with the idea that they rely more on network learning.

Conversely, the PAIRRE model is worse than the others for KGs with few relations. It is superior only for OGBL-WIKIKG2, indeed in the resampled metric it is the only model to convincingly beat RE. It is also affected by noise (for OGBL-WIKIKG2 at 0.5, MRR = 45.8/14.3), suggesting that all three mechanisms are in play.

5 Conclusions

We propose that KG learning is due to a combination of three mechanisms, namely, unstructured, network and motif learning, and presented results for synthetic and real KGs which illustrate the idea.

References

1. Abboud, R., Ceylan, İ.İ., Lukasiewicz, T., Salvatori, T.: BoxE: a box embedding model for knowledge base completion. [arXiv:2007.06267](https://arxiv.org/abs/2007.06267) [cs], July 2020
2. Akrami, F., Saeef, M.S., Zhang, Q., Hu, W., Li, C.: Realistic re-evaluation of knowledge graph completion methods: an experimental study. In: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, SIGMOD 2020, pp. 1995–2010. Association for Computing Machinery, New York (2020)
3. Allen, C., Balažević, I., Hospedales, T.M.: Interpreting knowledge graph relation representation from word embeddings. In: International Conference on Learning Representations (2021)
4. Alon, N., Krivelevich, M., Sudakov, B.: Finding a large hidden clique in a random graph. *Random Struct. Algorithms* **13**(3–4), 457–466 (1998)
5. Balažević, I., Allen, C., Hospedales, T.M.: TuckER: tensor factorization for knowledge graph completion. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 5184–5193 (2019). [arXiv: 1901.09590](https://arxiv.org/abs/1901.09590)
6. Bhattacharjee, R., Dasgupta, S.: What relations are reliably embeddable in Euclidean space? In: Algorithmic Learning Theory, pp. 174–195. PMLR (2020)
7. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
8. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 26, pp. 2787–2795. Curran Associates, Inc. (2013)
9. Chandrahas, A.S., Talukdar, P.: Towards understanding the geometry of knowledge graph embeddings. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 122–131. Association for Computational Linguistics, Melbourne, July 2018
10. Chao, L., He, J., Wang, T., Chu, W.: PairRE: knowledge graph embeddings via paired relation vectors. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, pp. 4360–4369 (2021)
11. Han, X., et al.: OpenKE: an open toolkit for knowledge embedding. In: EMNLP Proceedings (2018)
12. Heinonen, J.: Lectures on Analysis on Metric Spaces. Springer, New York (2012). <https://doi.org/10.1007/978-1-4613-0131-8>
13. Hu, W., et al.: Open graph benchmark: datasets for machine learning on graphs. [arXiv:2005.00687](https://arxiv.org/abs/2005.00687) [cs, stat], August 2020
14. Jain, P., Mausam, S.R., Chakrabarti, S.: Knowledge base completion: baseline strikes back (again) (2020)
15. Kadlec, R., Bajgar, O., Kleindienst, J.: Knowledge base completion: baselines strike back. In: Proceedings of the 2nd Workshop on Representation Learning for NLP, pp. 69–74. Association for Computational Linguistics, Vancouver, August 2017
16. Kilicoglu, H., Shin, D., Fiszman, M., Rosemblat, G., Rindflesch, T.C.: SemMedDB: a pubmed-scale repository of biomedical semantic predications. *Bioinformatics* **28**(23), 3158–3160 (2012)

17. Lehmann, J., et al.: DBpedia-a large-scale, multilingual knowledge base extracted from wikipedia. *Semant. web* **6**(2), 167–195 (2015)
18. Tanon, T.P., Weikum, G., Suchanek, F., et al.: Yago 4: a reason-able knowledge base. In: ESWC, pp. 583–596 (2020)
19. Pezeshkpour, P., Tian, Y., Singh, S.: Revisiting evaluation of knowledge base completion models. In: Automated Knowledge Base Construction (2020)
20. Ren, H., Hu, W., Leskovec, J.: Query2box: reasoning over knowledge graphs in vector space using box embeddings. [arXiv:2002.05969](https://arxiv.org/abs/2002.05969) [cs, stat], February 2020
21. Sun, Z., Deng, Z.-H., Nie, J.-Y., Tang, J.: RotatE: knowledge graph embedding by relational rotation in complex space. [arXiv:1902.10197](https://arxiv.org/abs/1902.10197) [cs, stat], February 2019
22. Sun, Z., Vashisht, S., Sanyal, S., Talukdar, P., Yang, Y.: A re-evaluation of knowledge graph completion methods. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 5516–5522. Association for Computational Linguistics, July 2020
23. Sutskever, I., Tenenbaum, J., Salakhutdinov, R.R.: Modelling relational data using bayesian clustered tensor factorization. In: Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., Culotta, A. (eds.) Advances in Neural Information Processing Systems, vol. 22 (2009)
24. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International Conference on Machine Learning (ICML) (2016)
25. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. [arXiv:1412.6575](https://arxiv.org/abs/1412.6575) [cs], August 2015

Biological Networks



CUBCO: Prediction of Protein Complexes Based on Min-cut Network Partitioning into Biclique Spanned Subgraphs

Sara Omranian^{1,2} and Zoran Nikoloski^{1,2(✉)}

¹ Bioinformatics, Institute of Biochemistry and Biology, University of Potsdam, 14476 Potsdam, Germany

Nikoloski@mpimp-golm.mpg.de

² Systems Biology and Mathematical Modeling, Max Planck Institute of Molecular Plant Physiology, 14476 Potsdam, Germany

Abstract. High-throughput approaches have generated large-scale protein-protein interaction (PPI) networks that are used in prediction of protein complexes. Here, we introduce CUBCO—a minimum cut-based algorithm that predicts protein complexes as biclique spanned subgraphs while relying on link prediction approaches to score and incorporate missing interactions. Our comprehensive analyses with PPIs from different organisms show that CUBCO performs on par with the best-performing approaches, that model protein complexes as biclique spanned subgraphs, and outperforms the remaining contenders. We also show that the usage of link prediction approaches in CUBCO improves the prediction of protein complexes on average 34.22% in all comparisons. Finally, CUBCO recovers ~40% and ~11% of known protein complexes from the Pan-Plant and Metazoan PPI networks. Therefore, CUBCO represents an efficient, parameter-free approach for accurate prediction of protein complexes from PPI networks.

Keywords: PPI networks · Protein complex prediction · Biclique spanned graph · Minimum cut

1 Introduction

Biological functions are supported by interactions between cellular components, including proteins. Protein-protein interaction (PPI) networks comprise nodes, corresponding to proteins, and edges, denoting protein interactions. The identification of protein complexes is important to understand the operational principles of cell systems, from signaling to metabolism. Several high-throughput technologies [1] have emerged to study PPIs, resulting in the assembly of large-scale PPI networks for well-studied, model organisms. However, due to the design and steps involved in these technologies, the resulting PPIs contain a sizeable portion of false-positive and false-negative interactions, which affect the identification of protein complexes [2].

Computational approaches are often used in the identification of protein complexes. These computational approaches have been grouped based on several criteria: (i) supervised *vs.* unsupervised, (ii) using only PPI network or also integrate other data [3].

To facilitate the comparison of these approaches and evaluate the predicted complexes, several gold standards, e.g. EcoCyc for *Escherichia coli* [4], MIPS, SGD, and CYC2008 for yeast [5–7], and CORUM for *H. sapiens* [8], have been assembled.

While there has been considerable improvement in the performance of computational approaches for prediction of protein complexes, there still some key challenges that remain open. The existing approaches rely on the idea that protein complexes correspond to highly connected clusters in PPI networks. As a result, they are often not capable of simultaneously identifying both dense and sparse, as well as, small and large protein complexes [3]. These approaches also depend on multiple parameters, which renders it difficult to interpret the resulting protein complexes. In contrast, the recently proposed PC2P and GCC-v [9, 10] represent parameter-free algorithms that model protein complex as biclique spanned subgraphs [11]. As a result, they can identify sparse as well as dense protein complexes independent of size, since biclique spanned subgraphs include stars, bicliques, and cliques as special graph classes. Nevertheless, if an approach for protein complex prediction uses only a given PPI network as input, its performance will be affected by erroneous and missing interactions in PPI networks. To overcome the issue of missing PPIs, link prediction algorithms have been recently proposed [12]. Therefore, the graph clustering and link prediction algorithms can be used jointly to improve protein complex prediction.

Building on this idea, here we introduce a new approach, referred to as CUBCO, that models protein complexes biclique spanned subgraphs, identified via minimum cut (unlike the local approaches in PC2P and GCC-v), and integrates link prediction to add most probable missing PPIs and investigate their effect on the performance of predicting protein complexes.

2 Results

2.1 CUBCO Predicts Protein Complexes Using Min Cut

Let $G = (V, E)$ be a graph with a set of nodes, V , corresponding to proteins, a set of edges, E , denoting PPIs. The simple graph denoted by $\overline{G} = (V, \{(u, v) | (u, v) \notin E\})$ denotes the complement of G . We formalize the concept of a protein complex by a biclique spanned graph, $G = (V, E)$, whose node set can be partitioned into two subsets, $V_1(G)$ and $V_2(G)$, and its edge set has the edges in the bipartite clique on $V_1(G)$ and $V_2(G)$ as a subset. Intuitively, a biclique spanned subgraph can be seen as a bipartite clique to which additional edges have been added. A biclique spanned graph has two properties: (i) the distance between any two nodes is at most two; (ii) Its complement, \overline{G} , is disconnected [13]. These properties provide a natural formalization of a network cluster based on connectedness, since the complement of a cluster defined this way is disconnected. Thereby, the goal is to partition the graph G , $C = \{C_1, C_2, \dots, C_k\}$, such that each C_i is a biclique spanned subgraph [11]. This can be obtained by utilizing not only local properties (e.g. second neighborhood or clustering coefficient [9, 10]), but also global properties of the graph.

The complement of graph, \overline{G} , contains edges that are not present in the original graph G . From biological perspective, the edges of \overline{G} include false-negative and

true-negative PPIs. Several studies have predicted the missing edges in PPI networks based on different concepts, with the network-based prediction of PPIs that relies on network walks of length three yielding the best results [12]. This approach favors inclusion of edges between nodes that are connected with a higher number of walks of length three in a given graph G . Here, we use the advantage of this approach, but rely on paths—instead of walks—of length three, to avoid effects of direct neighbors. Thereby, we assign weight to the edges of \overline{G} based on normalized number of paths of length three, given by:

$$w(u, v) = \sum_{i,j} \frac{P_{u,i} P_{i,j} P_{j,v}}{\sqrt{d_i d_j}}, \quad (1)$$

where $P_{i,j} = 1$ if nodes i and j are adjacent, and zero otherwise, and d_i denotes the degree of node i . Since one of the features of a biclique spanned subgraph is that its complement is disconnected, here we employ minimum cuts to discover biclique spanned subgraphs [14].

Given a graph G , CUBCO predicts protein complexes in three steps (Fig. 1): (i) determine \overline{G} , (ii) employ ideas from link prediction to weigh the edges in \overline{G} based on the degree-normalized number of path of length three between the end-nodes in G (Eq. (1), Algorithm 1), and (iii) iteratively identify minimum cuts in the edge-weighted graph \overline{G} (Algorithm 2), by using the Stoer-Wagner efficient, deterministic algorithm [15], until all resulting components are biclique spanned.

To render \overline{G} disconnected and obtain the biclique spanned subgraph, C_i , in G , we have to remove nodes, rather than edges (since edges in \overline{G} are not in G). Here, we used the min-cut algorithm that yields a partition of two node subsets, S_1 and S_2 , and a min-cut $E_{cut} = \{(u_i, v_i) | u_i \in S_1 \text{ and } v_i \in S_2, 1 \leq i \leq k\}$ where $k = |E_{cut}|$. Since, in practice, either S_1 or S_2 contain a single node, rather than considering all subsets for node removal, we consider the final biclique spanned subgraph to be either $C_1 = \{(S_1 \cup S_2) / \bigcup_{i=1}^k u_i\}$ or $C_2 = \{(S_1 \cup S_2) / \bigcup_{i=1}^k v_i\}$. The selection of the set C is guided by a score, Eq. (2), that shows the cohesiveness of the subgraph, $G[C]$, induced by C in G :

$$s(C) = \frac{|E_{in}(G[C])|}{|E_{out}(G[C])|}, \quad (2)$$

where $|E_{in}(G[C])|$ is the number of edges inside the subgraph, and $|E_{out}(G[C])|$ is the number of edges connecting the subgraph $G[C]$ to the rest of the network. CUBCO then selects C_i with the largest score and removes it from the graph \overline{G} . The complexity of CUBCO is $O(\frac{n^2}{d+1}(m + n \log n))$, where n is the number of nodes, m is the number of edges, and d is the minimum degree.

Algorithm 1 Preprocessing – Calculating edge weight in \bar{G}

```

1: procedure Preprocessing( $G$ )
2:    $\bar{G} \leftarrow complement(G)$ 
3:   for each  $edge(u, v)$  in  $\bar{G}$  do
4:      $w(u, v) \leftarrow$  normalized number of paths of length 3 in  $G$ 
5:     Update edge attribute of  $(u, v)$  in  $\bar{G}$ 
6:   return ( $\bar{G}$ )

```

Algorithm 2 CUBCO

```

1: procedure CUBCO( $\bar{G}$ )
2:   cluster_set  $\leftarrow []$ 
3:   While there is a node in  $\bar{G}$  do
4:      $(S_1, S_2) \leftarrow$  global min-cut Stoer-Wagner( $\bar{G}$ )
5:      $E_{cut} \leftarrow \{(u_i, v_i) | u_i \in S_1 \text{ and } v_i \in S_2, 1 \leq i \leq k\}$ 
6:      $C_i = \max(score((S_1 \cup S_2)/u_i), score((S_1 \cup S_2)/v_i))$ 
7:     Add  $C_i$  to cluster_set
8:     Remove  $C_i$  from  $\bar{G}$ 
9:   return (cluster_set )

```

2.2 Comparative Performance of CUBCO Without Link Prediction

We used twelve performance measures (Suppl. Information, see GitHub link in Method) to compare the predicted protein complexes from 17 approaches, including CUBCO, with all combinations of PPI networks and protein complexes from two *E. coli*, two *S. cerevisiae*, and one *H. sapiens* gold standards. The performance measures include: maximum matching ratio (MMR), fraction match (FRM), separation (SEP), positive predictive value (PPV), Sensitivity (SN), accuracy (ACC), precision, recall, F-measure, precision⁺, recall⁺, and F-measure⁺, with ranges between 0 and 1 and larger values indicating better performance (see Supplementary Information). We also calculated a composite scores given by the sum of MMR, FRM, ACC, and F-measure, that has been used in comparative analysis of approaches for protein complex prediction [16].

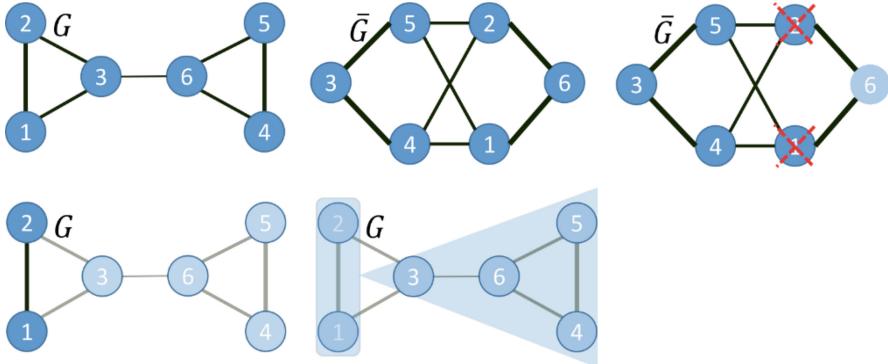


Fig. 1. Illustration of min-cut biclique spanned subgraph identification in CUBCO. The weight for each edge in the complement of the graph G is calculated based on Eq. (1). The global min-cut of the graph \bar{G} is obtained by Stoer-Wagner's algorithm. In this case, the algorithm separates node 6 from node-set $\{3, 4, 5\}$ by removing nodes 2 and 1. Hence, the first cluster $\{3, 4, 5, 6\}$ corresponds to a biclique spanned subgraph in G . After removal of this subgraph from G , the second cluster is given by $\{1, 2\}$.

For the combinations of Babu PPI networks and gold standards of *E. coli*, CUBCO resulted in the highest composite score (Fig. 2A, Tab. S3, Fig. S1, see GitHub link in Method). CUBCO exhibited composite score larger than half of the approaches for the combinations of Kong PPI network and gold standards of *E. coli* (Tab. S3, Fig. S1). For the combination of Gavin PPI network and the SGD gold standards in *S. cerevisiae* CUBCO exhibited the highest composite score in 62.5% of cases, preceded only by PC2P and GCC-v approaches that also represent protein complexes as biclique spanned subgraphs (Fig. 2B). In the remaining combinations of networks and gold standards CUBCO obtained a composite score higher than half of the compared approaches. More precisely, on average, the composite score of CUBCO is only 3.5% smaller than the composite score of the contenders ranked higher than CUBCO (Tab. S3, Fig. S1). In the case of *H. sapiens*, CUBCO exhibited the highest composite score among all other contenders for the PIPS PPI network and CORUM gold standard (Fig. 2C). However, for the combination of the STRING PPI network and CORUM, the composite score of CUBCO was on average 40.4% smaller than the contenders with a higher composite score (Tab. S3, Fig. S1).

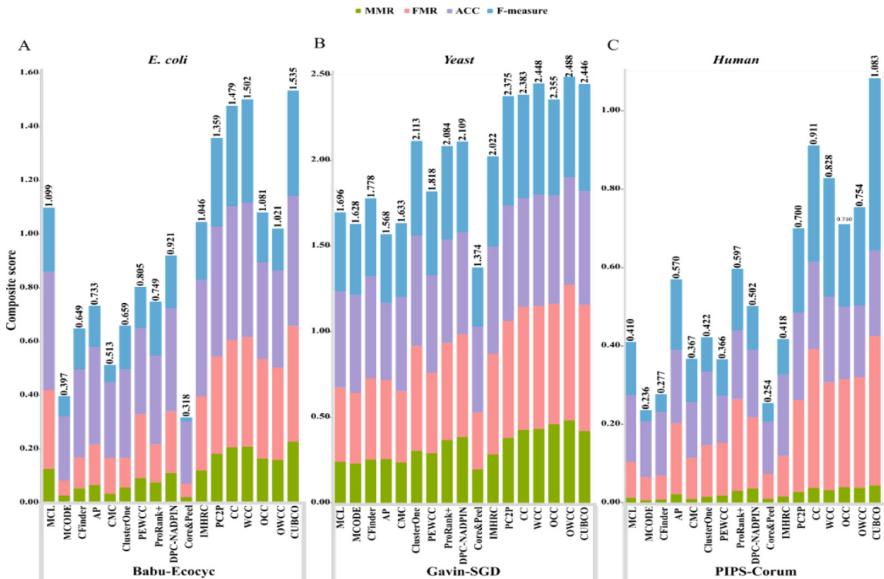


Fig. 2. Comparative analysis of approaches for prediction of protein complexes across PPI networks of different organisms. PPI networks for three organisms are considered (A) *E. coli*, (B) yeast, and (C) human. The comparative analyses are conducted with respect to a composite score that is a sum of MMR, FMR, ACC, and F-measure (see Supplementary file, Evaluation metric). Seventeen approaches, ordered by the year of publication, are compared on three PPI networks-gold standard combination. CUBCO outperforms all other approaches based on the composite score except in yeast, where it ranked third.

2.3 Integrating Link Prediction to Improve Performance of CUBCO

Next, we aimed to integrate a link prediction approach with CUBCO to resolve issues due to missing of PPIs from experimental approaches. To this end, we applied a modification of the L3 approach from [12], and ranked the missing edges by the number of paths of length 3 to avoid effects of immediate neighbors (Eq. (1)). We then selected the first 500, 1000, 1500, 2000, and 2500 of the ranked missing edges, and added them to the original network (as false-negative edges). Finally, we applied CUBCO on the modified network to predict the protein complexes. We found that the performance measures, and thereby, the composite score, increased slightly for all combinations of PPI networks and gold standards. The exception is the combinations of PPI networks of *E. coli* and KroganCore PPI network of yeast with Metabolic and SGD complexes, respectively. After careful investigation of these PPI networks, we realized that before inserting the new edges, they were of larger density and transitivity in comparison to all combinations of PPI networks of *E. coli* and KroganCore with Ecocyc and CYC2008 complexes. The result showed that the highest composite score for half of the combinations of PPI networks and gold standards was obtained by adding the first 500 ranked missing edges. For the combination of KroganExt PPI network of yeast with SGD complexes, the composite score improved gradually by

increasing the number of edges from 500 to 2500 (Fig. 3, Tabs. S4–6, Fig. S2). Overall, the finding implied that integrating of link prediction to graph clustering methods can enhance the predicted protein complexes.

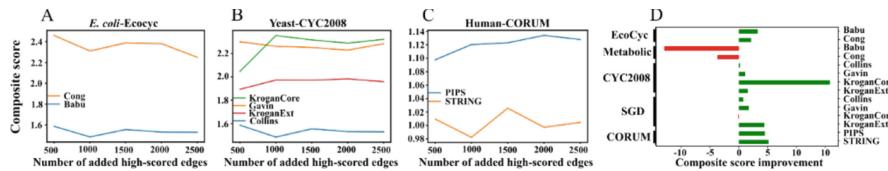


Fig. 3. Composite score due to integration of link prediction by adding the most probable edges. To investigate the composite score of CUBCO with the integration of link prediction, we inserted the first 500, 1000, 1500, 2000, and 2500 edges to the original PPI networks. The composite score is calculated for the combination of PPI networks and the gold standard of (A) *E. coli* and EcoCyc, (B) *S. cerevisiae* and CYC2008, and (C) *H. sapiens* and CORUM, respectively. On average, the composite score increases slightly with the increment of inserted edges across all PPI networks. (D) The original composite score of CUBCO across all combinations of organisms was compared with the obtained average composite score from inserting considered number of edges.

2.4 Prediction of New Protein Complexes in Conserved PPI Networks

We also applied CUBCO to predict new protein complexes in conserved PPI networks across 13 plant species, termed Pan-Plant [17], and across nine animal species, termed Metazoan [18] PPI network. For both PPI networks, we considered the high-confidence interactions with a score greater than 0.5 and used the gold standard, which is included in their corresponding studies (Tab. S1).

By applying CUBCO on the Pan-Plant PPI network, the resulting clusters fully coincided with 47 out of 118 known protein complexes (39.83%), and we predicted 249 new protein complexes that share no protein with the gold standards used (Tab. S7). To support the hypothesis that the proteins participating in a protein complex are involved in similar molecular functions and participate in the same cellular component and biological process, we determined the semantic similarity of the new protein complexes [19]. To this end, we employed the GOSim R package [20] to compute the median semantic similarity for every pair of proteins in each new protein complex. As a result, 77 of the predicted protein complexes (30.92%) showed GO semantic similarity of molecular functions equal to 1, indicating highest similarity of proteins function in the predicted protein complexes (Tab. S8). Further, we evaluated the predicted protein complexes by analyzing the domain-domain interactions (DDI) of proteins. We focused on the PPI network of *Arabidopsis thaliana*, identified the domains for ~46% of proteins based on the Pfam database [21], and considered the network of high-confidence (gold and silver) DDIs [22]. We found DDI support for 27 predicted protein complexes, of which 11 showed GO semantic similarity of 1 (Tab. S8). Finally, we only considered the predicted complexes that share no proteins with complexes in the gold standard, to obtain new predictions. We identified 165 new protein complexes, of which 19 have DDI support (Tab. S9). For instance, we found

that oxidative stress tolerance protein NQR (AT1G49670) forms a complex with AIM1 (AT4G29010) and ECHIA (AT4G16210). They share the same cellular compartment and are located in the peroxisome. One of the major functions of peroxisome is the Beta-oxidation of fatty acid [23], and AIM1 and ECHIA participate in this process, while NQR acts as a response to oxidative stress and co-regulated with Beta-oxidation genes [24]. Therefore, the predicted new complexes are of high quality and can form basis for future experimental confirmation.

Focusing on the Metazoan PPI network, CUBCO fully recovered 105 out of 981 known protein complexes (10.70%) and predicted 221 new protein complexes (Tab. S7). By conducting GO semantic similarity based on molecular function, 46 of the predicted protein complexes (20.81%) showed a value of 1 (Tab. S8). Based on the Pfam database, we then detected the domains for all proteins of the *H. sapiens* PPI network as part of the Metazoan network. We found that 88 of the predicted protein complexes have the DDI support, while only 15 of them also obtained GO semantic similarity of MF category equal to 1 (Tab. S8). Finally, we found only two new protein complexes that share no proteins from the gold standard, of which one is supported with DDI data (Tab. S9). The predicted complex with DDI support is the NXF1:NXT1 complex, which is also known as the TAP:p15 complex. This complex is a general mRNA nuclear export factor, and it is conserved from yeast to humans [25]. Altogether, the results indicate that CUBCO improves the performance of protein complex prediction and can find biologically meaningful protein complexes.

3 Method

3.1 Contending Approaches for Prediction of Protein Complexes

We compared the performance of CUBCO with sixteen other state-of-the-art methods, including: Markov Clustering (MCL), Molecular Complex Detection (MCODE), CFinder, Affinity Propagation (AP), Clustering-based on Maximal Cliques (CMC), Clustering with Overlapping Neighbourhood Extension (ClusterOne), PEWCC, Prorank⁺, Discovering Protein Complexes based on Neighbor Affinity and Dynamic Protein Interaction Network (DPC-NADPIN), Core&Peel, Inter Module Hub Removal Clustering (IMHRC), Protein Complexes from Coherent Partition (PC2P), and GCC-v (for references and software used, see Tab. S2). To facilitate fair comparison, the approaches are selected based on the public availability of their implementations and independence of any additional knowledge or data. Where applicable, we have used the default value of parameters.

3.2 PPI Networks and Gold Standards of Protein Complexes

We carried out all the experiments on PPI networks and gold standards of three model organisms: *E. coli*, *S. cerevisiae*, and *H. sapiens*. All the PPI networks are edge-weighted except one from *E. coli*. We used the two PPI networks of *E. coli* from [26, 27], named by the first authors' names (Babu and Cong). The two gold standards were given by manually curated protein complexes from Ecocyc [4] and protein

complexes based on the genome-scale metabolic network of *E. coli* [28]. For yeast, we used: Collins [29], Krogan core (edge-weight ≥ 0.273), Krogan extended (edge-weight ≥ 0.101) [30], and Gavin [31] PPI networks. The two gold standards were retrieved from CYC2008 [7] and complexes derived from the Saccharomyces Genome Database (SGD) [6]. For *H. sapeiens*, the two PPI networks were obtained from STRING (edge-weight ≥ 999) [32] and PIPS (edge-weight ≥ 25) [33]. In addition, we employed CORUM as the gold standard for human protein complexes [8] (Tab. S1). The comparative analysis of CUBCO and all the other approaches was performed on an Intel(R) Xeon(R) CPU E5-2670 v2 with 2.50GHz. CUBCO is freely available on GitHub at <https://github.com/SaraOmranian/CUBCO>, which also includes the supplementary figures and tables.

4 Conclusion

We proposed a new approaches, called CUBCO, to predict protein complexes from PPI networks. Since we have shown that partitioning a PPI network into biclique spanned subgraphs shows the best prediction performance of protein complexes [9, 10], CUBCO adopted the same concept. However, unlike the existing approaches, it relies on min-cuts, as a global network property, to determine the partition into biclique spanned subgraphs. CUBCO also employs paths of length three to rank false-negative interactions and include them in the networks to boost prediction performance. Future work will inspect the effect on coupling other approach for link prediction with approaches for prediction of protein complexes. In addition, we plan to consider weighted node cuts, rendering CUBCO applicable with large-scale proteomics and gene expression data.

References

- Peng, X., Wang, J., Peng, W., Wu, F.-X., Pan, Y.: Protein–protein interactions: detection, reliability assessment and applications. *Brief. Bioinform.* **18**(5), 798–819 (2016). p. bbw066
- Berger, B., Peng, J., Singh, M.: Computational solutions for omics data. *Nat. Rev. Genet.* **14**, 333–346 (2013)
- Wu, Z., Liao, Q., Liu, B.: A comprehensive review and evaluation of computational methods for identifying protein complexes from protein–protein interaction networks. *Brief. Bioinform.* **21**, 1531–1548 (2019)
- Keseler, I.M., et al.: The EcoCyc database: reflecting new knowledge about Escherichia coliK-12. *Nucleic Acids Res.* **45**, D543–D550 (2016)
- Mewes, H.W.: MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Res.* **32**, 41D – 44 (2004)
- Hong, E.L., et al.: Gene Ontology annotations at SGD: new data sources and annotation methods. *Nucleic Acids Res.* **36**, D577–D581 (2007)
- Pu, S., Wong, J., Turner, B., Cho, E., Wodak, S.J.: Up-to-date catalogues of yeast protein complexes. *Nucleic Acids Res.* **37**, 825–831 (2008)
- Giurgiu, M., et al.: CORUM: the comprehensive resource of mammalian protein complexes —2019. *Nucleic Acids Res.* **47**, D559–D563 (2018)

9. Omranian, S., Angeleska, A., Nikoloski, Z.: PC2P: parameter-free network-based prediction of protein complexes. *Bioinformatics* **37**, 73–81 (2021)
10. Omranian, S., Angeleska, A., Nikoloski, Z.: Efficient and accurate identification of protein complexes from protein-protein interaction networks based on the clustering coefficient. *Comput. Struct. Biotechnol. J.* **19**, 5255–5263 (2021)
11. Angeleska, A., Nikoloski, Z.: Coherent network partitions. *Discret. Appl. Math.* **266**, 283–290 (2019)
12. Kovács, I.A., et al.: Network-based prediction of protein interactions. *Nature Commun.* **10**, 3 (2019)
13. Akiyama, J., Harary, F.: A graph and its complement with specified properties. IV. Counting self-complementary blocks. *J. Graph Theory* **5**, 103–107 (1981)
14. Dantzig, G.B., Fulkerson, D.R.: 12. On the max-flow min-cut theorem of networks. In: *Linear Inequalities and Related Systems.* (AM-38), pp. 215–222. Princeton University Press (1957)
15. Stoer, M., Wagner, F.: A simple min cut algorithm. In: van Leeuwen, J. (ed.) *ESA 1994. LNCS*, vol. 855, pp. 141–147. Springer, Heidelberg (1994). <https://doi.org/10.1007/BFb0049404>
16. Nepusz, T., Yu, H., Paccanaro, A.: Detecting overlapping protein complexes in protein-protein interaction networks. *Nat. Methods* **9**, 471–472 (2012)
17. McWhite, C.D., et al.: A pan-plant protein complex map reveals deep conservation and novel assemblies. *Cell* **181**, 460–474.e14 (2020)
18. Wan, C., et al.: Panorama of ancient metazoan macromolecular complexes. *Nature* **525**, 339–344 (2015)
19. Cho, Y.-R., Hwang, W., Ramanathan, M., Zhang, A.: Semantic integration to identify overlapping functional modules in protein interaction networks. *BMC Bioinform.* **8**, 7 (2007)
20. Fröhlich, H., Speer, N., Poustka, A., Beißbarth, T.: GOSim – an R-package for computation of information theoretic GO similarities between terms and gene products. *BMC Bioinform.* **8**, 5 (2007)
21. Mistry, J., et al.: Pfam: The protein families database in 2021. *Nucleic Acids Res.* **49**, D412–D419 (2020)
22. Ziaeddine, A.S., Amina, A.-N., Hiba, N., Ritchie, D.W., Marie-Dominique, D.: PPIDomainMiner: inferring domain-domain interactions from multiple sources of protein-protein interactions, March 2021
23. Shani, N., Jimenez-Sanchez, G., Steel, G., Dean, M., Valle, D.: Identification of a fourth half ABC transporter in the human peroxisomal membrane. *Hum. Mol. Genet.* **6**, 1925–1931 (1997)
24. Wiszniewski, A.A.G., Zhou, W., Smith, S.M., Bussell, J.D.: Identification of two *Arabidopsis* genes encoding a peroxisomal oxidoreductase-like protein and an acyl-CoA synthetase-like protein that are required for responses to pro-auxins. *Plant Mol. Biol.* **69**, 503–515 (2008)
25. Aibara, S., Katahira, J., Valkov, E., Stewart, M.: The principal mRNA nuclear export factor NXF1:NXT1 forms a symmetric binding platform that facilitates export of retroviral CTE-RNA. *Nucleic Acids Res.* **43**, 1883–1893 (2015)
26. Babu, M., et al.: Global landscape of cell envelope protein complexes in *Escherichia coli*. *Nat. Biotechnol.* **36**, 103–112 (2017)
27. Cong, Q., Anishchenko, I., Ovchinnikov, S., Baker, D.: Protein interaction networks revealed by proteome coevolution. *Science* **365**, 185–189 (2019)
28. King, Z.A., et al.: BiGG models: a platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Res.* **44**, D515–D522 (2015)

29. Collins, S.R., et al.: Toward a comprehensive atlas of the physical interactome of *Saccharomyces cerevisiae*. *Mol. Cell. Proteom.* **6**, 439–450 (2007)
30. Krogan, N.J., et al.: Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*. *Nature* **440**, 637–643 (2006)
31. Gavin, A.-C., et al.: Proteome survey reveals modularity of the yeast cell machinery. *Nature* **440**, 631–636 (2006)
32. Szklarczyk, D., et al.: STRING v10: protein–protein interaction networks, integrated over the tree of life. *Nucleic Acids Res.* **43**, D447–D452 (2014)
33. McDowall, M.D., Scott, M.S., Barton, G.J.: PIPs: human protein-protein interaction prediction database. *Nucleic Acids Res.* **37**, D651–D656 (2009)



Quantifying Cellular Pluripotency and Pathway Robustness Through Forman-Ricci Curvature

Kevin A. Murgas¹(✉), Emil Saucan², and Romeil Sandhu^{1,3}

¹ Department of Biomedical Informatics, Stony Brook University,
Stony Brook, NY, USA

kevin.murgas@stonybrookmedicine.edu, romeil.sandhu@stonybrook.edu

² Department of Applied Mathematics, ORT Braude College, Karmiel, Israel
semil@braude.ac.il

³ Department of Applied Mathematics and Statistics, Stony Brook University,
Stony Brook, NY, USA

Abstract. In stem cell biology, cellular pluripotency describes the capacity of a given cell to differentiate into multiple cell types. From a statistical physics perspective, entropy provides a statistical measure of randomness and has been demonstrated as a way to quantitate pluripotency when considering biological gene networks. Furthermore, recent theoretical work has established a relationship between Ricci curvature (a geometric measure of “flatness”) and entropy (also related to robustness), which one can exploit to link the geometric quantity of curvature to the statistical quantity of entropy. Therefore, this study seeks to explore Ricci curvature in biological gene networks as a descriptor of pluripotency and robustness among gene pathways. Here, we investigate Forman-Ricci curvature, a combinatorial discretization of Ricci curvature, along with network entropy, to explore the relationship of the two quantities as they occur in gene networks. First, we demonstrate our approach on an experiment of stem cell gene expression data. As expected, we find Ricci curvature directly correlates with network entropy, suggesting Ricci curvature could serve as an indicator for cellular pluripotency much like entropy. Second, we measure Forman-Ricci curvature in a dataset of cancer and non-cancer cells from melanoma patients. We again find Ricci curvature is increased in the cancer state, reflecting increased pluripotency or “stemness”. Further, we locally examine curvature on the gene level to identify several genes and gene pathways with known relevance to melanoma. In turn, we conclude Forman-Ricci curvature provides valuable biological information related to pluripotency and pathway functionality. In particular, the advantages of this geometric approach are promising for extension to higher-order topological structures in order to represent more complex features of biological systems.

Keywords: Network geometry · Forman-Ricci curvature · Stem cell biology · Cancer biology

1 Introduction

A major biological phenomenon is cellular differentiation, whereby pluripotent stem cells undergo phenotypic evolution to approach one of many possible differentiated states. Understanding how cellular dynamics shape the trajectories of differentiation can provide us with a deeper knowledge of the processes that govern phenotypes, which could allow us to predict cellular behavior in various biological processes and ultimately may inform our strategy for influencing these complex biological systems with targeted pharmacology. For example, recent work in this regard have ranged from stem cell biology [1–4], cancer biology [1,2], cellular reprogramming [5], to defining biological robustness and drug resistance [6–8]. Epigenetic evolution is familiarly motivated by C.H. Waddington [9] as a hill-like “landscape”, in which cellular state is intrinsically coupled to a quantity similar to potential energy in classical mechanics (Fig. 1A). That is, cellular differentiation could be modeled as a ball rolling down the hillside of this landscape, gaining momentum and following “grooves” towards a state of lower potential energy and perhaps settling into a local minima. However, the mathematical quantification of this landscape in regards to cellular trajectories and pluripotency in general has only been recently investigated [3,4,10] and to a large part, remains an open problem.

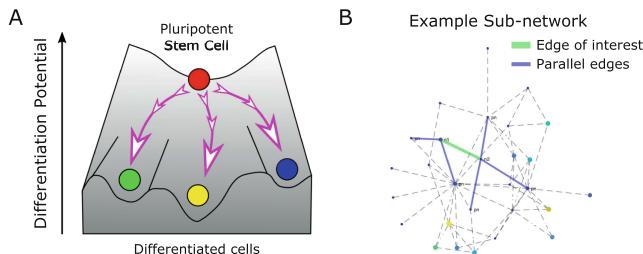


Fig. 1. A: Stem cell differentiation as a landscape, wherein stem cells move down a gradient of potential energy to reach local minima at unique differentiated cell states. B: Example sub-network of PPI network, with an edge of interest (green) and corresponding parallel edges (blue) labelled.

To this end, in order to estimate and capture the complex dynamics of cellular differentiation, one can model the underlying biological systems as a network. While a complete review of such network modeling is beyond the scope of this work [11], this study focuses on protein-protein interaction (PPI) networks [12,13]. Here, individual genes are treated as components in a large network of interacting proteins, wherein relationships such as protein binding or enzyme catalysis are modeled as corresponding edges between nodes on a graph and for which such “interactions” have been validated by experimental studies. In this regard, many recent studies have applied PPI networks to model the differentiation landscape [1,2,14]. More importantly, several of these recent works

attempt to mathematically quantify the general notion of pluripotency from a statistical mechanics perspective, in which the differentiation capacity of a cell can be described by entropy measured on the PPI network. In particular, [14] utilizes a notion of graph entropy to examine the “randomness” of gene expression data overlaid onto a PPI network. These studies demonstrate gene network entropy to be a powerful statistical descriptor of cell pluripotency, decreasing upon cellular differentiation and increasing in situations such as cancer [1,2].

While statistical mechanics and in particular notions of entropy are well-known with ubiquitous applications beyond cellular biology, recent work has shown that Ricci curvature (a geometric measure of “flatness”) may also be reinterpreted as a statistical quantity and proxy for entropy [15–17]. This is particularly fascinating in the continuous case as one is able to unify a purely deterministic geometric (local) quantity of curvature with that of a statistical (equilibrium) quantity of entropy. From the biological perspective, this provides a “bridge” to investigate geometric (and topological) properties of biological networks to better capture cellular functionality in facets not generally realizable from a purely classical statistical lens [18]. That said, discretizing the continuous definition of Ricci curvature (and other forms of curvature in Riemannian geometry [19]) is not necessarily straightforward especially over a discrete metric space (e.g., graph) where differential operators are not readily provided. As such, several works have focused on discretizing Ricci curvature for graphs including the definitions of Ollivier [20] and Forman [22]. While both discretize Ricci curvature for graphs, Ollivier-Ricci curvature [20] is motivated through the statistical idea that probabilistic neighborhoods are closer (farther) than their centers depending on positive (negative) curvature, using optimal transport theory to compute such a distance, whereas Forman-Ricci curvature provides an explicit combinatorial curvature formula based on Bochner’s method of decomposing the combinatorial Laplacian (more detail in Sect. 2) [22–25]. While both of these measures work well on 1-dimensional simplicial complexes, i.e. graphs with only nodes and undirected edges, Forman-Ricci curvature is in fact formulated for higher order structures such as faces, which are composed of multiple nodes and edges [26]. Therefore, in the context of biological gene networks where multiple genes can interact simultaneously in common pathways, Forman-Ricci curvature may prove advantageous over previous applications of Ollivier-Ricci curvature from both computational and mathematical perspectives.

Of importance, theoretical work [15–17] has demonstrated a relationship between Boltzmann entropy and Ricci curvature (i.e. changes in entropy are positively correlated with changes in curvature). Several works have exploited this concept in a variety of venues from cancer biology [18,27,28], neuroscience [29,30], wireless network congestion [31], economics [32], to graph-based Ricci flows [25,33]. Here, most of the biological work has focused on Ollivier’s definition. However, there exist intrinsic computational and mathematical limitations to the generalization of this definition to the higher order topological structures necessary to capture biological functionality underlying cellular dynamics. Therefore, our study examines curvature and entropy in the context

of stem cell differentiation and cancer with the goal of establishing a direct link between the deterministic quantity of Forman-Ricci curvature and the statistical quantity of network entropy. Ultimately, we aim to explore the capability of Forman-Ricci curvature as a meaningful descriptor of cellular pluripotency and pathway functionality. If curvature analysis can indicate which genes are associated with pluripotency and how these genes drive pluripotency, this would suggest geometric theory could guide applications in biology and medicine to exert dynamic control over cellular differentiation processes and pathologic processes as in cancer. Perhaps more importantly, this will lay the foundation for an exploration of higher-dimensional biological structures through extension of Forman's definition.

This paper is organized as follows: In Sect. 2, we provide preliminaries and necessary background. In Sect. 3, we describe our computational approach. Following this, Sect. 4 presents results that highlight Forman-Ricci curvature as a descriptor of pluripotency in a stem cell dataset and pathway functionality in a melanoma dataset. Lastly, Sect. 5 presents conclusions and discussions of future directions.

2 Background

2.1 Motivation: Interplay of Entropy and Curvature

Critical to our motivation for this study are several works exploring the relationship of entropy and curvature in geometric spaces [15–17]. The core of the approach of these studies involves the dependence of various geometric quantities on the lower bound of Ricci curvature. Namely, for a given Riemannian manifold, the existence of a lower quadratic bound on Ricci curvature, $Ric(M) \geq \kappa I$ where I is the identity matrix, implies K-convexity of Boltzmann entropy on the same manifold [17]. This property can be exploited to establish a positive correlation between changes in entropy ΔS and changes in curvature ΔRic ,

$$\Delta S \times \Delta Ric \geq 0. \quad (1)$$

An additional key relationship to these two variables involves robustness, which can be defined as the rate of return to equilibrium in a dynamical system. The Fluctuation theorem [27, 34] asserts a positive correlation between changes in entropy ΔS and changes in robustness ΔR , which therefore, by Eq. (1), implies the same relationship between Ricci curvature and robustness,

$$\Delta S \times \Delta R \geq 0 \implies \Delta Ric \times \Delta R \geq 0. \quad (2)$$

These findings provide direction for this study, serving as the theoretical groundwork for exploring Ricci curvature in gene networks as a statistical indicator for pluripotency and pathway robustness. Given that global network entropy has been demonstrated as a descriptor of cellular pluripotency [1, 2, 14], we aim to extend Ricci curvature as a geometric quantity on biological gene networks. Then, we can apply the relationships of curvature to entropy and robustness, respectively, to assess Ricci curvature as a potential descriptor of pluripotency (globally) and pathway robustness (locally).

2.2 Biological Graphs: Topological and Geometric Information

In the context of biology, graphs such as protein-protein interaction networks can represent the complex system of interactions between diverse proteins. On one hand, topological information alone can provide valuable information about biological networks [35]. Nevertheless, gene expression array data can be overlaid onto these graphs to augment the underlying topology with geometric node and edge weights, providing richer and more complex analysis than merely topological information. In this paper, we seek to analyze such gene networks from a geometric perspective.

Previous studies examining gene network entropy utilize a formulation based on Shannon entropy [14]. In brief, local entropy S_i at a node i can be determined based on the interaction probabilities p_{ij} of neighboring nodes, and a subsequent global entropy rate SR can be computed as a weighted-average of local entropies based on the stationary (invariant) distribution π of the network,

$$S_i = \sum_{j \in N_i} p_{ij} \log p_{ij} \quad SR = \sum_{i=1}^n \pi_i S_i. \quad (3)$$

By realizing a gene network as a geometric object, we aim to exploit the relationship of entropy and Ricci curvature, shown in Eq. (1), to pivot from the statistical quantity of entropy into the geometric quantity of curvature. We also seek to utilize the relationship of robustness and Ricci curvature, in Eq. (2), by examining Ricci curvature within local network components, which here would represent individual genes and gene pathways, to quantify changes in functional robustness that could indicate critical pathways involved in biological processes such as tumorigenesis.

Importantly, the discrete structure of gene networks requires a discretization of Ricci curvature, for which we select Forman-Ricci curvature due to its combinatorial nature, computational advantages, and adaptability to higher-order structures.

2.3 Forman-Ricci Curvature

In the discrete setting, Forman-Ricci curvature can be developed through a combinatorial analogue to the Bochner-Weitzenböck decomposition of the Riemannian (or Hodge [21]) Laplacian \square_p into a rough (or Bochner) Laplacian B_p and a combinatorial curvature operator F_p [22],

$$\square_p = B_p + F_p. \quad (4)$$

This definition applies to CW-complexes, a general class of topological structures that includes graphs. In this sense, a graph is considered a complex consisting of nodes (0-cells) and edges (1-cells) glued together at their boundaries, i.e. their nodes. For each edge in the graph, we use Forman's approach to derive an explicit combinatorial curvature formula that depends only on the weights of the edge, the edge end-nodes, and neighboring (parallel) edges [22].

Let us consider an arbitrary graph consisting of nodes and edges (Fig. 1B). For a given edge e with ascribed edge weight w_e , coupled with end-nodes v_1 and v_2 with respective node weights w_{v_1} and w_{v_2} , Forman-Ricci curvature is defined as

$$\mathbf{F}(e) = w_e \left(\frac{w_{v_1}}{w_e} - \sum_{e_{v_1} \sim e} \frac{w_{v_1}}{\sqrt{w_e w_{e_{v_1}}}} + \frac{w_{v_2}}{w_e} - \sum_{e_{v_2} \sim e} \frac{w_{v_2}}{\sqrt{w_e w_{e_{v_2}}}} \right), \quad (5)$$

where $e_{v_i} \sim e$ denotes parallel edges that share a common node v_i with edge e .

This definition applies to both undirected and directed graphs [24]. In the case of a *directed* graph, the same definition is applied, with the caveat that only parallel edges concordant in direction with edge e are considered and any opposite edge $\overline{v_2 v_1}$ is disregarded. Notably, as we consider here only simple graphs of nodes (0-cells) and edges (1-cells), we disregard higher order structures such as faces (2-cells), which do appear in the full derivation [22].

As Forman-Ricci curvature is defined on each edge of the graph, one can define a contracted nodal curvature as the mean of all edge curvatures incident to a given node,

$$\mathbf{F}(v) = \frac{1}{\deg(v)} \sum_{e_v \sim v} \mathbf{F}(e_v). \quad (6)$$

In the case of a *directed* graph, nodal curvature is defined as the mean of all incoming edge curvatures minus all outgoing curvatures,

$$\mathbf{F}(v) = \frac{1}{\deg_{IN}(v)} \sum_{e_{v_{IN}} \sim v} \mathbf{F}(e_{v_{IN}}) - \frac{1}{\deg_{OUT}(v)} \sum_{e_{v_{OUT}} \sim v} \mathbf{F}(e_{v_{OUT}}). \quad (7)$$

A global average of curvature \mathbf{F}_{GA} can be computed as a weighted average of nodal curvatures, weighted by the stationary distribution π of the graph,

$$\mathbf{F}_{GA} = \sum_{i=1}^n \pi_i \mathbf{F}(v_i). \quad (8)$$

3 Approach

3.1 Gene Expression Pre-processing to Graph Construction

In this note, we regard an RNA-sequencing data set as a matrix with a row for each unique gene measured and a column for each unique sample in the experiment. This data is typically provided as integer read counts that correspond to the number of times an observed RNA sequence read was aligned to a specific gene transcript. These read count data were pre-processed by first performing quantile normalization to reduce bias between samples, then a \log_2 transformation was applied, which brings large values closer to the distribution and allows for similar variation across different magnitudes. The goal of these pre-processing steps was to reduce systematic bias and improve the analysis of biologically meaningful data [36].

Protein-protein interaction (PPI) graph topology was defined based on protein interaction data from Pathway Commons (<https://www.pathwaycommons.org>) [13]. These interaction data were further processed by a sparsification technique described in [1], which removed likely false-positive edges. As such, we directly utilized the same interaction network as [1,2], downloaded from the SCENT Github repository (<https://github.com/aet21/SCENT>, file: data/net13Jun12.Rda). This PPI network contained 8434 nodes corresponding to unique genes and 303,600 edges describing protein-protein interactions between pairs of nodes.

From here, a (pre-processed) gene expression matrix D measuring expression of M genes across N samples (i.e. D is a matrix of dimension $M \times N$) was overlaid onto the PPI network. First, all nodes corresponding to genes not measured in the gene expression dataset were removed and subsequently only the largest strongly connected component of the resulting network was considered. Next, for each individual sample k , taking the corresponding column vector \mathbb{R}_+^M of gene expression, $E := D_{\cdot k}$ (i.e., $E \in \mathbb{R}_+^M$), node weight was directly defined as the (pre-processed) gene expression value,

$$w(v_i) = E_i. \quad (9)$$

Edge weights were then defined based on the mass-action law [14,37,38], which states the rate of interaction is directly proportional to the concentration (or expression) of each gene,

$$w(e_{ij}) = E_i E_j. \quad (10)$$

We note that across multiple samples, the same PPI topology was used and new node and edge weights were defined based on the gene expression profile of a given sample.

3.2 Gene Network Curvature Analysis

For each individual sample, gene expression was overlaid onto the PPI network and node weights and edge weights were defined as above. A stochastic matrix P was created by normalizing edge weights at each node by the sum of outgoing edge weights, such that the sum of outgoing edge weights in P sum to 1 for any node,

$$P_{ij} = w(e_{ij}) / \sum_{\forall k \in N(i)} w(e_{ik}). \quad (11)$$

This matrix P is notably similar to the transition matrix of a Markov chain, in which the transition probabilities are proportional the expression level of neighboring nodes. The stationary distribution π of the graph, which represents the equilibrium of a random walk on the graph and satisfies $\pi = P\pi$ (i.e. π is an eigenvector of the stochastic matrix P with an eigenvalue of 1) was taken as the normalized first eigenvector of P .

Using the normalized edge weights defined by P , we applied the combinatorial Forman-Ricci curvature formula, in Eq. (5), to compute curvature at each edge. Then, edge curvature values were used to compute nodal curvature values using Eq. (7). A global average of curvature was computed for each sample, using the stationary distribution π to calculate a weighted average over the nodal curvatures, as in Eq. (8). To compare entropy to curvature, the stochastic matrix P was used to compute local Shannon entropy at each node and subsequent global entropy rate by Eq. (3).

For local curvature analysis, we develop a differential curvature analysis based on the principles of differential expression analysis [39]. For each individual gene, local curvature values across all samples within a group (i.e. cancer or non-cancer samples) were compared to define a log₂ fold-change in curvature ($\text{log}_2\text{FC} = \log_2(\text{mean}_2/\text{mean}_1)$) and statistically assessed using an unpaired t-test, with the resulting p-values adjusted for multiple comparisons using the Benjamini-Hochberg step-up procedure [40] to produce false discovery rates (FDR, or q-values). To select genes with significant changes in curvature among the groups, we apply a cutoff of $q < 0.05$ and absolute $\text{log}_2\text{FC} > 2$. These genes with differential curvature were fed into Reactome pathway analysis [41], which utilizes a hypergeometric test to determine pathway overrepresentation, to produce a list of pathways that were retained if the Reactome FDR was $q < 0.05$.

4 Results

We now present results on few experiments that illustrate the viability of utilizing gene network Forman-Ricci curvature as an indicator for cellular pluripotency as well as pathway robustness. The aim of this results section is to provide proof-of-concept and is by no means a complete analysis. This said, the results herein lay a foundation for future work, in which we aim to examine higher dimensional

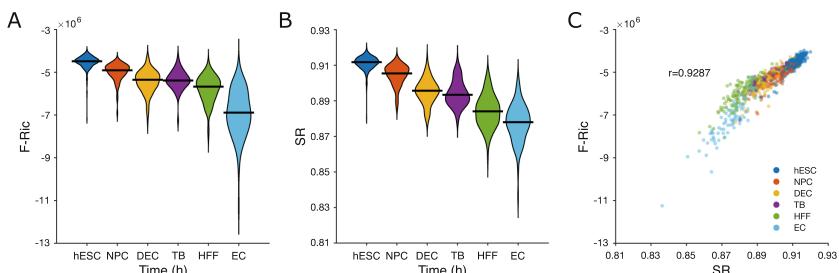


Fig. 2. Stem-cell differentiation. A: Violin plot showing distributions of global average curvature in differentiated cell types. Horizontal black bars are medians. A trend of decreasing curvature with increasingly differentiated cell types is observed. B: Same as A but for global entropy. A similar trend of decreasing entropy is observed. C: Scatter plot of global weighted-average nodal curvature vs global entropy. The two quantities were strongly correlated at $r = 0.9287$ (Pearson).

structures that more accurately capture biological differentiation dynamics. As such, for the purposes of demonstrating our approach of calculating Forman-Ricci curvature on biological gene networks, we focused our analyses on two previously published single-cell RNA sequencing datasets, first from a stem cell differentiation study [42], and second from a melanoma study including cancer and non-cancer cells [43]. This data is publicly available through the NCBI GEO portal (<https://www.ncbi.nlm.nih.gov/geo>) under accession numbers GSE75748 and GSE72056, respectively.

4.1 Differentiation Cell Type (GSE75748)

We focused on a subset of the stem cell dataset examining a cell-type experiment, which measured gene expression on $n = 1018$ single cells from 6 different cell types representing increasingly differentiated cells derived from undifferentiated human embryonic stem cells [42]. Cell-type abbreviations: *hESC* human embryonic stem cell, *NPC* neuronal progenitor cell, *DEC* definitive endoderm cell, *TB* trophoblast-like cell, *HFF* human foreskin fibroblast, *EC* endothelial cell.

We assessed gene network Forman-Ricci curvature and entropy for each individual single-cell sample (Fig. 2). We observed an overall decrease in curvature in more differentiated cells compared to stem and progenitor cells. In agreement with previous studies examining network entropy on this same data [2], a similar trend was observed for entropy. Notably, we observed curvature on these samples to exhibit strongly negative values, which indicated the local edge curvatures on average were predominantly negative, whereas entropy values were normalized to the interval $[0, 1]$. We emphasize the magnitudes of individual curvature values are not nearly as important as the *changes* in curvature between cell types as these changes reflect relative differences in pluripotency. Pearson correlation between curvature and entropy values was 0.9287, suggesting a strong relationship of the two values as expected from the theoretical correlation of entropy and curvature.

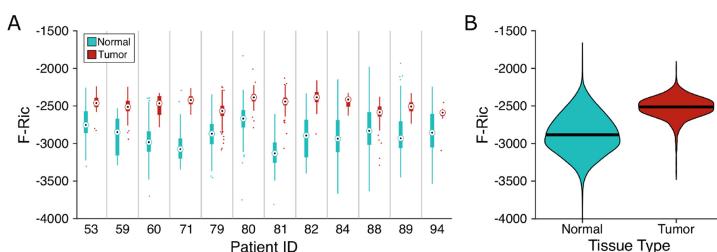


Fig. 3. Melanoma. A: Box plot of global average curvature distributions for normal and tumor cells of individual patients. An increase in curvature among tumor cells relative to normal cells was observed in each patient. B: Violin plot of global average curvature in normal and tumor cells across all patients. Horizontal black bars are medians.

4.2 Melanoma (GSE72056)

The melanoma dataset contained $n = 4513$ total single-cells from 19 melanoma patients, of which $n_N = 3256$ were non-cancer (normal) cells and $n_T = 1257$ were cancer (tumor) cells [43].

We again computed Forman-Ricci curvature for each single-cell sample (Fig. 3), observing a clear trend of increased curvature in cancer cells compared to non-cancer cells within each patient and across all patients. These findings indicate increased cellular pluripotency or “stemness” in cancer relative to non-cancer tissue.

We extended this analysis to examine the local curvature at each node (gene), specifically applying a differential curvature analysis approach in order to identify genes with changing curvature between cancer and non-cancer in these melanoma cases. We identified 210 genes with significantly increased curvature and 250 genes with decreased curvature. Subsequently, we utilized Reactome pathway analysis [41] to determine gene pathways to which these increasing and decreasing curvature genes correspond. We found 13 increased curvature pathways including the CCT/TriC protein folding pathway, FGFR4 pathway (oncogene in melanoma [44]), and matrix remodeling pathways, each of which having been implicated in various aspects of melanoma pathogenesis [44–46]. We also found 22 decreased curvature pathways including the FGFR2 pathway (tumor suppressor in melanoma [44]) and several immune pathways, which would typically have protective effects against melanoma [47,48]. Therefore, the observations of increased curvature in oncogenic pathways and decreased curvature in tumor suppressive pathways highlights the relationship of Ricci curvature and robustness, suggesting that cancer may exhibit increased curvature and thus robustness in pathways beneficial to the cancer. These results demonstrate the capacity of Ricci curvature to identify gene pathway functionality in pathologic processes such as cancer.

5 Conclusions

In this work, we have presented Forman-Ricci curvature as a discrete geometric quantity to evaluate biological gene networks. We were motivated by previous work examining gene network entropy, which found global entropy to be a potential descriptor of differentiation potential [1,2,14]. We extended theoretical findings linking entropy and Ricci curvature to establish a relationship between the statistical quantity of entropy and geometric quantity of Ricci curvature [15–17]. In accomplishing this, we selected Forman-Ricci curvature as a discretization of Ricci curvature, owing to its flexibility in assessing more general, high-order biological graph structures. Our findings indicate that Ricci curvature decreases upon cellular differentiation and increases in cancer, similarly to entropy. While network entropy has been previously shown to reflect cellular pluripotency, to our knowledge this is the first example of Ricci curvature representing the same information. Additionally, we demonstrate local analysis of Ricci curvature on a melanoma dataset to reveal several gene pathways with known relevance to

melanoma, which exhibits the capacity of Ricci curvature to reveal changes in robustness related to pathway functionality in pathologic processes such as cancer.

While this approach is promising, there are caveats. Many specific details in the gene network analysis are subject to consideration, such as the pre-processing technique, the PPI network topology, and the definitions of node and edge weights. Future directions of this work should include extending the approach to higher-dimensional graph structures, in particular to 2-dimensional simplicial complexes [26], as well as exploring geometric flows such as Ricci flow on biological networks [25]. These avenues could reveal additional biological meaning in regards to cellular pluripotency and differentiation trajectories.

Acknowledgements. K.M. and R.S. were supported by the National Science Foundation grant ECCS-1749937, the U.S. Air Force Office of Scientific Research grant FA9550-18-1-0130, and E.S. was supported by the German-Israeli Foundation grant I-1514-304.6/2019.

References

1. Banerji, C.R., et al.: Cellular network entropy as the energy potential in Waddington's differentiation landscape. *Sci. Rep.* **3**(1), 1–17 (2013)
2. Teschendorff, A.E., Enver, T.: Single-cell entropy for accurate estimation of differentiation potency from a cell's transcriptome. *Nat. Commun.* **8**(1), 1–15 (2017)
3. Schiebinger, G., et al.: Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell* **176**(4), 928–943 (2019)
4. Weinreb, C., Rodriguez-Fraticelli, A., Camargo, F.D., Klein, A.M.: Lineage tracing on transcriptional landscapes links state to fate during differentiation. *Science* **367**(6479), eaaw3381 (2020)
5. MacArthur, B.D., Ma'ayan, A., Lemischka, I.E.: Systems biology of stem cell fate and cellular reprogramming. *Nat. Rev. Mol. Cell Biol.* **10**(10), 672–681 (2009)
6. Kitano, H.: Towards a theory of biological robustness. *Mol. Syst. Biol.* **3**, 137 (2007)
7. Kitano, H.: The theory of biological robustness and its implication in cancer. In: Bringmann, P., Butcher, E.C., Parry, G., Weiss, B. (eds.) *Systems Biology. SCHER-ING FOUND*, vol. 61, pp. 69–88. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-31339-7_4
8. Amit, I., Wides, R., Yarden, Y.: Evolvable signaling networks of receptor tyrosine kinases: relevance of robustness to malignancy and to cancer therapy. *Mol. Syst. Biol.* **3**(1), 151 (2007)
9. Waddington, C.H.: *The Strategy of Genes*. Allen & Unwin, London (1957)
10. Wang, J., Zhang, K., Xu, L., Wang, E.: Quantifying the Waddington landscape and biological paths for development and differentiation. *Proc. Natl. Acad. Sci.* **20**, 8257–8262 (2011)
11. Wang, Y.R., Huang, H.: Review on statistical methods for gene network reconstruction using expression data. *J. Theor. Biol.* **362**, 53–61 (2014)
12. Klingström, T., Plewczynski, D.: Protein-protein interaction and pathway databases, a graphical review. *Brief. Bioinform.* **12**(6), 702–713 (2011)

13. Rodchenkov, I., et al.: Pathway Commons 2019 Update: integration, analysis and exploration of pathway data. *Nucleic Acids Res.* **48**(D1), D489–497 (2020)
14. Teschendorff, A.E., Sollich, P., Kuehn, R.: Signalling entropy: a novel network-theoretical framework for systems analysis and interpretation of functional omic data. *Methods* **67**(3), 282–293 (2014)
15. von Renesse, M.K., Sturm, K.T.: Transport inequalities, gradient estimates, entropy and Ricci curvature. *Commun. Pure Appl. Math.* **58**(7), 923–940 (2005)
16. Sturm, K.T.: On the geometry of metric measure spaces. *Acta Math.* **196**(1), 65–131 (2006)
17. Lott, J., Villani, C.: Ricci curvature for metric-measure spaces via optimal transport. *Ann. Math.* **169**(3), 903–991 (2009)
18. Pouryahya, M., Oh, J.H., Mathews, J.C., Deasy, J.O., Tannenbaum, A.R.: Characterizing cancer drug response and biological correlates: a geometric network approach. *Sci. Rep.* **8**(1), 1–2 (2018)
19. do Carmo, M.P.: *Riemannian Geometry*. Birkhäuser, Boston (1992)
20. Ollivier, Y.: Ricci curvature of Markov chains on metric spaces. *J. Funct. Anal.* **256**(3), 810–864 (2009)
21. Hodge, W.V.D.: *The Theory and Applications of Harmonic Integrals*, 2nd edn. Cambridge University Press, Cambridge (1952)
22. Forman, R.: Bochner’s method for cell complexes and combinatorial Ricci curvature. *Discrete Comput. Geom.* **29**(3), 323–374 (2003). <https://doi.org/10.1007/s00454-002-0743-x>
23. Sreejith, R.P., Mohanraj, K., Jost, J., Saucan, E., Samal, A.: Forman curvature for complex networks. *J. Stat. Mech: Theory Exp.* **2016**(6), 063206 (2016)
24. Saucan, E., Sreejith, R.P., Vivek-Ananth, R.P., Jost, J., Samal, A.: Discrete Ricci curvatures for directed networks. *Chaos, Solitons Fractals* **118**, 347–360 (2019)
25. Weber, M., Saucan, E., Jost, J.: Characterizing complex networks with Forman-Ricci curvature and associated geometric flows. *J. Complex Netw.* **5**(4), 527–550 (2017)
26. Saucan, E., Weber, M.: Forman’s Ricci curvature - from networks to hypernetworks. In: Aiello, L.M., Cherifi, C., Cherifi, H., Lambiotte, R., Lió, P., Rocha, L.M. (eds.) *COMPLEX NETWORKS 2018*. SCI, vol. 812, pp. 706–717. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05411-3_56
27. Sandhu, R., et al.: Graph curvature for differentiating cancer networks. *Sci. Rep.* **5**(1), 1–13 (2015)
28. Sandhu, R., Tannenbaum, S., Georgiou, T., Tannenbaum, A.: Geometry of correlation networks for studying the biology of cancer. In: Proceedings of the IEEE 55th Conference on Decision and Control (CDC), December 2016, pp. 2501–2506 (2016)
29. Weber, M., Stelzer, J., Saucan, E., Naitsat, A., Lohmann, G., Jost, J.: Curvature-based methods for brain network analysis. arXiv preprint [arXiv:1707.00180](https://arxiv.org/abs/1707.00180), July 2017
30. Farooq, H., Chen, Y., Georgiou, T.T., Tannenbaum, A., Lenglet, C.: Network curvature as a hallmark of brain structural connectivity. *Nat. Commun.* **10**(1), 1–11 (2019)
31. Wang, C., Jonckheere, E., Banirazi, R.: Wireless network capacity versus Ollivier-Ricci curvature under Heat-Diffusion (HD) protocol. In: Proceedings of the IEEE 2014 American Control Conference, pp. 3536–3541 (2014)
32. Sandhu, R., Georgiou, T., Tannenbaum, A.: Market fragility, systemic risk, and Ricci curvature. arXiv preprint [arXiv:1505.05182](https://arxiv.org/abs/1505.05182), May 2015

33. Sandhu, R., Liu, J.: Maxwell's demon: controlling entropy via discrete Ricci flow over networks. In: Masuda, N., Goh, K.-I., Jia, T., Yamanoi, J., Sayama, H. (eds.) NetSci-X 2020. SPC, pp. 127–138. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-38965-9_9
34. Demetrius, L.A.: Boltzmann, Darwin and directionality theory. *Phys. Rep.* **530**(1), 1–85 (2013)
35. Siegal, M.L., Promislow, D.E., Bergman, A.: Functional and evolutionary inference in gene networks: does topology matter? *Genetica* **129**(1), 83–103 (2007). <https://doi.org/10.1007/s10709-006-0035-0>
36. Liu, X., et al.: Normalization methods for the analysis of unbalanced transcriptome data: a review. *Front. Bioeng. Biotechnol.* **7**, 358 (2019)
37. Guldberg, C.M., Waage, P.: Über die chemische Affinität. *J. Prakt. Chem.* **19**(69), 13 (1879)
38. Schreiber, G.: Protein-protein interaction interfaces and their functional implications. In: Protein-Protein Interaction Regulators, pp. 1–24. Royal Society of Chemistry, December 2020
39. Costa-Silva, J., Domingues, D., Lopes, F.M.: RNA-Seq differential expression analysis: an extended review and a software tool. *PLoS ONE* **12**(12), e0190152 (2017)
40. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Stat. Soc. B* **57**(1), 289–300 (1995)
41. Jassal, B., et al.: The reactome pathway knowledgebase. *Nucleic Acids Res.* **48**(D1), D498–D503 (2020)
42. Chu, L.F., et al.: Single-cell RNA-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome Biol.* **17**(1), 1–20 (2016). <https://doi.org/10.1186/s13059-016-1033-x>
43. Tirosh, I., et al.: Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science* **352**(6282), 189–196 (2016)
44. Czyz, M.: Fibroblast growth factor receptor signaling in skin cancers. *Cells* **8**(6), 540 (2019)
45. Boudiaf-Benmammar, C., Cresteil, T., Melki, R.: The cytosolic chaperonin CCT/TRiC and cancer cell proliferation. *PLoS ONE* **8**(4), e60895 (2013)
46. Napoli, S., et al.: Functional roles of matrix metalloproteinases and their inhibitors in melanoma. *Cells* **9**(5), 1151 (2020)
47. Passarelli, A., Mannavola, F., Stucci, L.S., Tucci, M., Silvestris, F.: Immune system and melanoma biology: a balance between immunosurveillance and immune escape. *Oncotarget* **8**(62), 106132–106142 (2017)
48. Tucci, M., et al.: Immune system evasion as hallmark of melanoma progression: the role of dendritic cells. *Front. Oncol.* **9**, 1148 (2019)



Emergence of Stable Functional Cliques in Developing Neural Networks

Myles Akin and Yixin Guo^(✉)

Drexel University, Philadelphia, PA 19104, USA
yixin.guo@drexel.edu

Abstract. Complex networks constructed from neural correlations exhibit nonrandom structures hypothesized to be related to neural information processing. These networks can be characterized by topological features such as hubs, communities and small world connectivity. A particular network structure of interest is a clique - a fully connected subgraph - which may indicate the existence of underlying neural ensembles. We introduce a multilayer network method to observe the persistence of functional cliques over population bursts as well as their development as the underlying neural network forms connections. Using data from developing cultures on MEAs, we show that cliques become more numerous and persistent over population bursts as neural cultures age. These results provide evidence for the formation of neural ensembles in cultured neural networks.

Keywords: Computational neuroscience · Multiplex networks · Cliques

1 Introduction

Abstract networks have found effective use in studying the organizational properties of neural systems [6, 14, 15, 19, 21]. Typically, these networks are constructed to represent either anatomical connectivity (structural networks), or inferred statistical dependency within recordings of neural activity (functional networks). Multiple scales of neural tissues have been studied using networks; from membrane recordings of individual neurons to fMRI studies of brain regions. Simulation studies have shown that there may be a significant relationship between an inferred functional network and its underlying structural network [1, 2, 7, 10].

In this paper, we investigate the emergence of cliques in developing dissociated neural populations recorded using microelectrode arrays (MEA). Electrodes showing high correlation may record from neurons within the same structural circuit [14]. We study correlations within population burst to determine functional structures. Population bursts are short periods of time, usually between 100–1000 ms, during which a large number of neurons in the population show bursting behavior. Bursts of spikes are thought to be the primary way that information is passed and synapses are strengthened between neurons, as in spike timing dependent plasticity (STDP) [9, 19, 24]. By constructing a multiplex network

where each layer corresponds the functional connectivity of a population burst, defined in the next section, we can develop measures to find persistent (temporally stable) cliques. These cliques may provide evidence for the emergence of neural ensembles, groups of neurons that show spatiotemporal co-activation either through direct or indirect synaptic connectivity [12, 23]. That is, electrodes in a clique showing relatively high persistence may be recording from neurons in a neural ensemble. Neural ensembles are of particular interest due to their hypothesized involvement in memory formation [4, 12, 14, 19, 23].

During the development of neural cultures on MEAs, we expect early weak synaptic connections to result in few cliques as well as cliques being active in fewer population bursts. As the neural population strengthens synaptic connections, we may expect to see more cliques in population bursts as well as specific cliques being present in multiple populations bursts. By observing the emergence of persistent clique activation across multiplex networks as cultures mature, we can provide evidence for the self-organization of neural populations into neural ensembles.

2 MEA Data, Population Bursts and Correlations

Microelectrode arrays (MEAs) are a popular tool for studying neural cultures [11–13, 22]. Populations of neurons or neural tissue are placed on the array of electrodes, which record the electrical activity of neurons within a small distance. MEAs allow experimentalists to study spiking activity of small populations of intact neural networks from tissue slices [17, 18, 20] or the development of dissociated neural populations [5, 11, 12, 16, 22]. While the synaptic structure of the neural population is unknown, the recorded spike trains can be used to construct functional networks which may infer some general topological properties of the underlying structural network. The use of functional networks constructed from MEA recordings have resulted in the discovery of many interesting network features. Studies of cortical slices cultured on MEA's have shown the existence of hub, cluster and community structures [18, 20]. Dissociated neural populations have been shown to evolve rich-club and small-world structure when allowed to develop on MEAs [5, 16].

We use spike time data sets from two separate preparations of dissociated embryonic rat cortical neural populations, made publicly available by Wagenaar, et al. [22]. The first dataset (group 1) consists of four cultures from a preparations with recordings on days 12, 18 and 24. A second dataset (group 2) consists of four additional cultures with recordings on days 14, 21, 28 and 35. Each MEA consists of 59 recording electrodes. Note that each group had one MEA whose recording varied by a day, however, as we are looking for trends, we do not believe data from slightly different days affects our analysis.

A spike time series consists of the timestamps a channel (electrode) records spikes. For our purposes, we convert the spike time series into binned spike trains. A spike train for electrode i is a binary vector, $\vec{x}_i[t]$ with $i \in \{1, 2, 3, \dots, 59\}$. Each entry of $\vec{x}_i[t]$ represents a bin of time, where $\vec{x}_i[t_n] = 1$ if the electrode records

a spike in time bin t_n and zero otherwise. We use 1 ms time bins for burst identification.

We identify population bursts in a similar manner as in Wagenaar et al. [22]. To start, we consider each channel spike train independently and look for patterns of ≥ 4 spikes in a 100 ms sliding window. The window begins at $t = 0$ and slides 1 ms at each iteration. Windows that have ≥ 4 spikes and have temporal overlap are combined to form a *burstlet window*. The first and last spikes in this burstlet window then determine the starting point and ending point for our *burstlet*.

If more than 15 electrodes have burstlets with temporal overlap, we consider that to be a population burst. To determine the beginning, t_b , and end, t_e , of the population burst, we take the minimal and maximal spike times over all burstlets in that population burst. This process prevents chopping long duration population bursts into multiple population bursts and ignores population bursts of short duration. We thus obtain a set of population burst boundary times, denoted by $P = \{(t_b^\alpha, t_e^\alpha) : \alpha \in \{1, 2, 3, \dots, L\}\}$, for each culture and day with L population bursts. We denote a single population burst P^α . An example of a few identified population bursts is shown in Fig. 1.

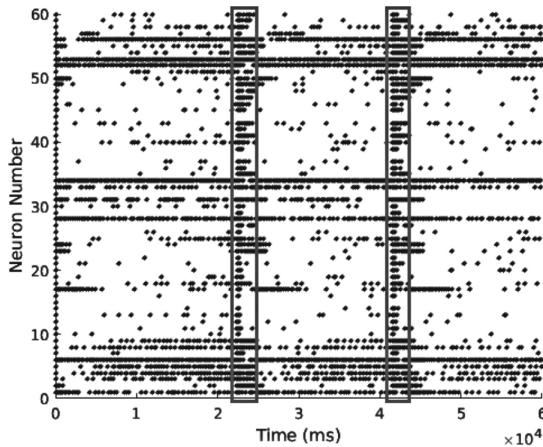


Fig. 1. Raster plot of short section MEA data with the the electrode number on the vertical axis and time on the horizontal. Identified population bursts are outlined in the rectangular boxes.

After identifying the population burst for each culture and day, we aim to find the correlation between the activity of all 59 electrodes. A common method for comparing the similarity of binary vectors is the Jaccard index. The Jaccard index gives a measure of pairwise correlations by calculating the ratio of the number of bins both vectors are 1 to the sum of the number of times either or

both are 1. That is, for two binary vectors \vec{x}_i^α and \vec{x}_j^α in burst α , we find the Jaccard index as follows

$$(M_{ij}^\alpha)^{11} = \text{number of times when, } \vec{x}_i^\alpha[t] = \vec{x}_j^\alpha[t] = 1 \quad (1)$$

$$(M_{ij}^\alpha)^{10} = \text{number of times when, } \vec{x}_i^\alpha[t] = 1 \text{ and } \vec{x}_j^\alpha[t] = 0 \quad (2)$$

$$(M_{ij}^\alpha)^{01} = \text{number of times when, } \vec{x}_i^\alpha[t] = 0 \text{ and } \vec{x}_j^\alpha[t] = 1 \quad (3)$$

$$J_{ij}^\alpha = \frac{(M_{ij}^\alpha)^{11}}{(M_{ij}^\alpha)^{11} + (M_{ij}^\alpha)^{10} + (M_{ij}^\alpha)^{01}} \quad (4)$$

Where i and j are electrodes and α is the population burst index. This index is symmetric, $J_{ij}^\alpha = J_{ji}^\alpha$. We can treat this J_{ij}^α as a correlation matrix with entries between 0 and 1.

For a given population burst P^α , the binned spike trains bounded by t_b^α and t_e^α give binary vectors for each electrode. When the spiking activity recorded by electrodes i and j are similar, $(M_{ij}^\alpha)^{11}$ is relatively large compared to $(M_{ij}^\alpha)^{10}$ and $(M_{ij}^\alpha)^{01}$, resulting in a larger Jaccard index. Therefore, we can find pairs of electrodes with similar activity. We use 4 ms time bins to account for delays in spike generation.

3 Monoplex and Multiplex Networks

For each population burst P^α , we construct a monoplex functional network, $G^\alpha = (V^\alpha, E^\alpha)$, which consists of finite sets of vertices V^α and edges E^α . In our case, V^α is the set of all 59 electrodes. The set of edges is given by $E^\alpha = \{\{v_i^\alpha, v_j^\alpha\} : v_i^\alpha, v_j^\alpha \in V^\alpha\}$ where $\{v_i^\alpha, v_j^\alpha\}$ is an edge connecting vertices v_i^α and v_j^α . The edges for single layer networks are determined by thresholding the Jaccard correlation matrices. We use a linear combination of a local threshold, μ_{ij}^α and a global threshold μ . A low global threshold is used to eliminate spurious correlations. The local threshold is as follows: for entry J_{ij}^α , the local threshold level is given by $\mu_i^\alpha = \frac{1}{59} \sum_j J_{ij}^\alpha$. An edge $e_{ij}^\alpha = \{v_i^\alpha, v_j^\alpha\}$ exists based on the following rule:

$$e_{ij}^\alpha \begin{cases} \in E^\alpha & \text{if } J_{ij}^\alpha \geq \mu_i^\alpha + \mu \text{ and } J_{ij}^\alpha \geq \mu_j^\alpha + \mu \\ \notin E^\alpha & \text{otherwise} \end{cases} \quad (5)$$

This set of edges e_{ij}^α gives the network structure of population burst P^α .

We construct a multilayer network $M = (\mathcal{G}, \mathcal{E})$ consisting of a set of monoplex networks $\mathcal{G} = \{G^1, G^2, \dots, G^L\}$ and a set of interlayer edges \mathcal{E} that connect vertices in one layer to vertices in a different layer. We set up our multilayer networks such that each layer corresponds to a given population burst (see previous section). A particular kind of multilayer network in which the vertex sets in each layer are the same is called a multiplex network. This type network is ideal for our situation since each layer has 59 vertices corresponding to the MEA electrodes (these don't change between population bursts). Formally, we can define a multiplex network as follows:

Definition 1. Multiplex Network. Let $\mathcal{G} = \{G^\alpha : \alpha \in \{1, 2, \dots, L\}\}$ be a set of L single layer networks, then $M = (\mathcal{G}, \mathcal{E})$ is a multiplex network if

$$G^\alpha = (V^\alpha, E^\alpha), \text{ where for all } \alpha, \beta \in \{1, 2, \dots, L\}, V^\alpha = V^\beta \quad (6)$$

$$\mathcal{E} = \{E^{\alpha\beta} \subseteq V^\alpha \times V^\beta : \alpha, \beta \in \{1, 2, \dots, L\}, \alpha \neq \beta\} \quad (7)$$

$$E^{\alpha\beta} = \{\{v_i^\alpha, v_i^\beta\} : v_i^\alpha \in V^\alpha, v_i^\beta \in V^\beta, \beta = \alpha + 1\} \quad (8)$$

that is $E^{\alpha\beta}$ are interlayer edges connecting corresponding vertices in subsequent layers only.

As we only use interlayer edges to maintain vertex correspondence, all these edges exist with values set to one.

4 Multiplex Network Measures

In this section, we develop two measures to study the emergence of cliques and their temporal stability (persistence) in these functional multiplex networks.

A clique with k vertices (electrode) is an induced subgraph that is isomorphic to the fully connected graph, K_k with k vertices. For a graph $G = (V, E)$, an induced subgraph is a set of vertices in V along with all edges in E that have those vertices as endpoints. In a graph with N vertices, there are $\binom{N}{k}$ possible sets of k vertices, of which any could be a clique. We denote a subset of vertices with cardinality k by U_q^k , where $q \in \Omega = \{1, 2, 3, \dots, \binom{N}{k}\}$ is an arbitrary labeling of the possible sets; for our case $N = 59$ for all layers. Functional cliques are sets of electrodes whose recorded spiking activity are all correlated above threshold.

4.1 Average Number of k -Cliques

The number of cliques that we observe in a multiplex population burst network help us determine, in general, how coordinated spiking activity is on that day. At the early stages of culture development, when synaptic connectivity isn't settled, we may expect to see little to no activity correlation within population bursts giving very few cliques. As ensembles form, we expect a significant increase in the presence of cliques within population bursts. The average number of k -cliques observed gives us an idea of the strength of local connectivity in populations bursts on a given day. This measure is given formally by the following definition.

Definition 2. Let $M = (\mathcal{G}, \mathcal{E})$ with L layers and $\mathcal{G} = \{G^\alpha : \alpha \in \{1, 2, 3, \dots, L\}\}$. For a $q \in \Omega$, let $(I_q^k)^\alpha$ be the induced subgraph of U_q^k in layer α . Let $\Xi_k^\alpha = \{U_q^k : \text{for any } q \in \Omega, (I_q^k)^\alpha \simeq K_k\}$ be the set of k cliques in layer α . Here, \simeq denotes isomorphic. The average number of k -cliques in a multiplex network then is

$$\bar{\Xi}_k = \frac{1}{L} \sum_{i=1}^L |\Xi_k^\alpha|$$

where $||$ denotes the cardinality.

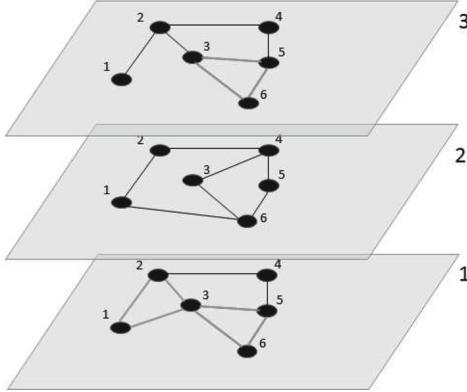


Fig. 2. Three layer multiplex network. There are two sets of 3 vertices that form cliques $U_1^3 = \{1, 2, 3\}$ and $U_2^3 = \{3, 5, 6\}$. The layer supports are $\delta_1^3 = \{1\}$ and $\delta_2^3 = \{1, 3\}$. Therefore the 3-clique persistence of these are $\eta_1^3 = \frac{1}{3}$ and $\eta_2^3 = \frac{2}{3}$.

An example of this measure is shown in Fig. 2 for an a small multiplex network. For the first layer in this example, two vertex sets from cliques, therefore $\Xi_3^1 = \{U_1^3, U_2^3\}$. In layer three, $\Xi_3^3 = \{U_2^3\}$. There are no vertex sets in layer two that form cliques. Therefore the average number of 3-clique is $\bar{\Xi}_3 = 1$.

With this we find the average number of k -cliques per layer for a given day. A significant increase in this measure as the cultures age indicates that the neurons may be undergoing self-organization into ensembles with strong local connectivity.

4.2 k -Clique Persistence

We may expect that electrodes recording from neural ensembles would form a clique in multiple population bursts, due to the temporal aspect of co-activation. Therefore, observing sets of vertices that are correlated enough to form cliques in multiple layers provide evidence of neural ensemble formation. We can consider this the persistence of a clique consisting of a given set of vertices. On the other hand, cliques that are observed in very few layers may emerge from random activation of neural circuits and not as the result of neural ensemble formation. Here we develop a network measure for the study of the persistence of cliques in a multiplex network.

Again let $(I_q^k)^\alpha$ be the induced subgraph of U_q^k with size k in layer α and K_k the complete graph with k vertices. We use the following definition

Definition 3. Let $M = (\mathcal{G}, \mathcal{E})$ be a multiplex network with L layers. The layer support, δ_q^k , of vertex set U_q^k is, for a given $q \in \Omega$, the set of layers $\delta_q^k = \{\alpha \in \{1, 2, \dots, L\} : (I_q^k)^\alpha \simeq K_k\}$.

That is, δ_q^k is the set of layers for which U_q^k is a clique. Using this definition, we can now define k -clique persistence.

Definition 4. The k -clique persistence of U_q^k , for a $q \in \Omega$, is given by

$$\eta_q^k = \frac{|\delta_q^k|}{L}$$

This measure gives the fraction of layers that the vertex set U_q^k forms a clique. This indicates that $\eta_q^k \in \{\frac{1}{L}, \frac{2}{L}, \dots, \frac{L-1}{L}, 1\}$. In general, we will be more interested in the count of vertex sets of size k with a specific k -clique persistence η^k , such that $\eta^k \in \{\frac{1}{L}, \frac{2}{L}, \dots, \frac{L-1}{L}, 1\}$. We define a function $H(\eta^k)$ as follows

$$H(\eta^k) = \sum_{q \in \Omega} I_{\eta_q^k = \eta^k}$$

where $I_{\eta_q^k = \eta^k}$ is the indicator function equal to 1 when $\eta_q^k = \eta^k$ and 0 otherwise. This function provides a count of the number of sets of vertices with η^k persistence. Observing H for all possible η^k gives a count distribution for the possible persistence values. This distribution of k -cliques persistence gives a more complete picture of the spatiotemporal aspects of clique behavior.

5 Results

The number of population bursts identified in cultures varied widely over days, between ≈ 15 – 150 . To accommodate this wide variation, we present comparisons between days using the average of a window of 15 layers (populations bursts) for analysis. That is, we obtain network measures for first 15 layers, then iterating by 5 to obtain network measures for layers 5–20, and so on. The average of the measures over each window of 15 layers is then found for each culture on each day and used for analysis. We provide results here for cliques of size $k = 2, 3$, and 4, note that 2-cliques are edges. For all results presented, we use global threshold $\mu = 0.08$. We use Friedman's nonparametric ANOVA and Dunn's post hoc to test for statistical significance.

5.1 Average Number of k -Cliques, $k = 2, 3, 4$

As cultures mature, increased local connectivity may give rise to an increase in the number of functional cliques observed in population bursts.

Figure 3 shows the distribution of $\bar{\Xi}_k$ for both groups (group one (a), group two (b)) on the respective days in vitro with $k = 2, 3, 4$. We observe a significant ($p < 0.1$) increase in the number of cliques for all k considered with post-hoc analysis showing statistically significant difference between the first and last days for both groups.

Further, we observe that on the earliest day for which we have data, the number of 2-cliques exceeds the number of 3- and 4-cliques. By the second day of recording, the average number of 3- and 4-cliques exceeds the number of 2-cliques. This provides evidence that neural circuits are spatially increasing in

size. Further, as the number of 4-cliques surpasses, on average, the number of 3-cliques on days beyond the first, we conclude that there must exist many bursts with cliques $k > 4$.

Group two shows slightly different behavior than group one as it matures. Day 21 has fairly high variance for the average number of 3, 4-cliques and has a small dip in the average number of cliques (all k) on day 28. The reason for this is unclear, however, the overall behavior still indicates a similar pattern of statistically significant clique (neural circuit) formation.

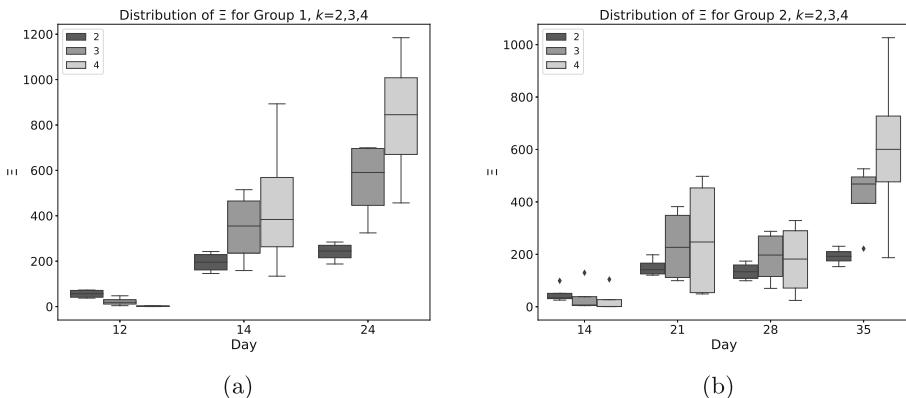


Fig. 3. Distribution of the average number of 2-cliques (edges) for (a) group 1 and (b) group 2. The average number of edges increases from early stages of development to later stages. We note that group 2 appears to form functional edges slower than group 1.

5.2 k -Clique Persistence for $k = 2, 3, 4$

The previous section provided evidence for increased spatial co-activation of neural circuits as the culture matured through observing an increase in the average number of k -cliques ($k = 2, 3, 4$). Here, we use the persistence measure developed in Sect. 4.2 to investigate the temporal stability of circuit co-activation (i.e. cliques) across population bursts.

We first examine the persistence on 3-cliques in a representative culture. Figure 4 gives log-log plot of the distribution, $H(\eta^3)$, for our representative culture on three different days of development (days 12, 18, 24). The shape of these distributions (reminiscent of power-law distributions) indicate that the vast majority of vertex (electrode) sets of size three, U_q^3 , do not show correlated activity within any, or at least very few, population bursts. While this behavior is observed for all days considered, we note that as the culture matures, the number of cliques with correlated activity across multiple population bursts increases. In Fig. 4, day 12 has a maximum persistence of $\frac{5}{12}$ single set of vertices. That is, this set of electrodes recorded correlated enough activity to form a clique in 5

of the 15 population bursts (layers). For day 24, the maximum persistence was $\frac{12}{15}$ for 3 vertex sets. In addition to an increase in the maximal persistence, we observe that as the culture ages, a greater number of cliques show higher levels of persistence. For instance, the number of vertex sets showing a persistence of $\frac{5}{15}$ is 4 on day 18 while it is over 100 by day 24. This behavior was seen with clique $k = 1, 4$ as well.

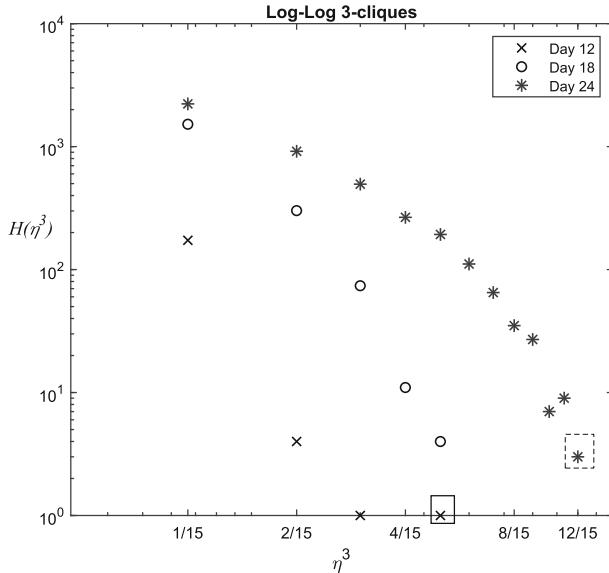


Fig. 4. Log-Log plot of the distribution of persistence of 3-cliques, $H(\eta^3)$, for a representative MEA. The distributions show a somewhat power law distribution. The maximal persistence for day 12, $\eta = \frac{5}{15}$, is enclosed in the solid box while the maximal persistence for day 24, $\eta^3 = \frac{12}{15}$, is enclosed in the dashed box.

To study the statistical significance of persistence increase, we develop a measure that provides an indication of the overall persistence on a culture on a given day. We can then use this measure to determine whether there was a statistically significant increase in persistence over culture maturation. Define the function $\phi(\eta^k)$ as follows

$$\phi(\eta^k) = \begin{cases} \ln H(\eta^k) & H(\eta^k) > 0 \\ 0 & \text{otherwise} \end{cases}$$

Then we define the measure Λ as

$$\Lambda = \sum_{\eta^k} \eta^k \phi(\eta^k)$$

This function places high weight on cliques with relatively high persistence and low weight on cliques with relatively low persistence. This prevents the much larger number of low-persistence cliques from dominating the measure. We can now determine if there is a significant increase in this value Λ as the cultures age.

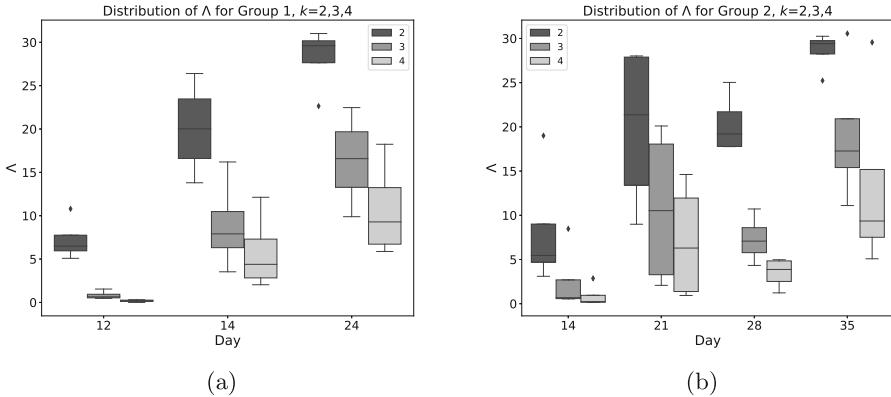


Fig. 5. Distribution of Λ with $k = 2, 3, 4$ for both group 1 (a) and group 2 (b). We see a significant increase in this measure for both groups and clique sizes as the cultures age.

The distributions of Λ for both groups 1 and 2 on all days and $k = 2, 3, 4$ are shown in Fig. 5. All k -cliques show a significant increase in the persistence measure ($p < 0.1$). Post hoc analysis indicates that for all k , group 1 has a significant difference between days 12 and 24 and group 2 between days 14 and 35. The figures also show that values of Λ drop as the clique size increases. The reduction in the mean value of Λ indicates that as clique size increases, cliques become less persistent despite being more numerous (previous section). This may be due to larger scale circuits being more prone to noise than small circuits. Group 2 shows unexpected behavior on day 21 where the variation in the data is very high, as we saw with average number of cliques. This may indicate issues during the plating process, further experimental studies would be needed to confirm.

6 Discussion

In this paper, we developed a novel method to study the emergence of functional cliques in population bursts using multiplex network. We constructed our multiplex networks such that the network structure of each layer corresponded to the functional connectivity of a population bursts. With these networks, we analyzed the local structure, in the form of k -cliques, over multiple layers using

two different network measures. We observed a significant increase in the average number of cliques and the persistence of cliques within population bursts as neural populations age. These results provide evidence that the neural populations are undergoing self-organization into neural ensembles. These ensembles may provide ready circuits for the formation of memories and for the processing of information.

An extension to this study will look at the similarity of clique structure between layers. This type of measure may indicate if cliques are consistently active together; this is, if two cliques exist in multiple population bursts, the similarity between these population bursts will increase. An increase in the similarity between populations bursts would indicate that certain neural ensembles may be driving population burst activity. We may also see periodic behavior in similarity, indicating that different groups of ensembles may be communicating and activating together.

References

1. Akin, M., Onderdonk, A., Dzakpasu, R., Guo, Y.: Functional reconstruction of dyadic and triadic subgraphs in spiking neural networks. In: Cherifi, H., Gaito, S., Quattrociocchi, W., Sala, A. (eds.) COMPLEX NETWORKS 2016. SCI, vol. 693, pp. 697–708. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-50901-3_55
2. Akin, M., Onderdonk, A., Guo, Y.: Effects of local network topology on the functional reconstruction of spiking neural network models. *Appl. Netw. Sci.* **2**(1), 1–22 (2017). <https://doi.org/10.1007/s41109-017-0044-1>
3. Boccaletti, S., et al.: The structure and dynamics of multilayer networks. *Phys. Rep.* **544**, 1–122 (2014)
4. Brunel, N.: Is cortical connectivity optimized for storing information? *Nat. Neurosci.* **19**(5), 749–755 (2016)
5. Downes, J.H., et al.: Emergence of a small-world functional network in cultured neurons. *PLoS Comput. Biol.* **8**, e1002522 (2012)
6. Frolov, N.S., Maksimenko, V.A., Khramova, M.V., Pisarchik, A.N., Hramov, A.E.: Dynamics of functional connectivity in multilayer cortical brain network during sensory information processing. *Eur. Phys. J. Spec. Top.* **228**(11), 2381–2389 (2019). <https://doi.org/10.1140/epjst/e2019-900077-7>
7. Ito, S., Hansen, M.E., Heiland, R., Lumsdaine, A., Litke, A.M., Beggs, J.M.: Extending transfer entropy improves identification of effective connectivity in a spiking cortical network model. *PLoS ONE* **6**, e27431 (2011)
8. Ito, S., et al.: Large-scale, high-resolution multielectrode-array recording depicts functional network differences of cortical and hippocampal cultures. *PLoS ONE* **9**(8), e105324 (2014)
9. Izhikevich, E.M.: *Dynamical Systems in Neuroscience*. MIT Press, Cambridge (2007)
10. Kobayashi, R., Kitano, K.: Impact of network topology on inference of synaptic connectivity from multi-neuronal spike data simulated by a large-scale cortical network mode. *J. Comput. Neurosci.* **35**, 109–124 (2013). <https://doi.org/10.1007/s10827-013-0443-y>

11. Massobrio, P., Pasquale, V., Martinoia, S.: Self-organized criticality in cortical assemblies occurs in concurrent scale-free and small-world networks. *Sci. Rep.* **5**, 10578 (2015)
12. Niedringhaus, M., Chen, X., Conant, K., Dzakpasu, R.: Synaptic potentiation facilitates memory-like attractor dynamics in cultured in vitro hippocampal networks. *PLoS ONE* **8**(3), e57144 (2013)
13. Pasquale, V., Massobrio, P., Bologna, L.L., Chiappalone, M., Martinoia, S.: Self-organization and neuronal avalanches in networks of dissociated cortical neurons. *Neuroscience* **153**(4), 1354–1369 (2008)
14. Poli, D., Pastore, V.P., Massobrio, P.: Functional connectivity in in vitro neuronal assemblies. *Front. Neural Circ.* **9**, 57 (2015)
15. Rubinov, M., Sporns, O.: Complex network measures of brain connectivity: uses and interpretations. *Neuroimage* **52**, 1059–1069 (2011)
16. Schroeter, M.S., Charlesworth, P., Kitzbichler, M.G., Paulsen, O., Bullmore, E.T.: Emergence of rich-club topology and coordinated dynamics in development of hippocampal functional networks in vitro. *J. Neurosci.* **35**(14), 5459–5470 (2015)
17. Segev, R., Baruchi, I., Hulata, E., Ben-Jacob, E.: Hidden neuronal correlations in cultured networks
18. Shimono, M., Beggs, J.M.: Functional clusters, hubs, and communities in the cortical microconnectome. *Cereb. Cortex* **25**, 3743–3757 (2015)
19. Sporns, O.: Networks of the Brain. MIT Press, Cambridge (2010). *Phys. Rev. Lett.* **92**, 118102 (2004)
20. Timme, N.M., et al.: High-degree neurons feed cortical computations. *PLoS Comput. Biol.* **12**, e1004858 (2016)
21. Vaiana, M., Muldoon, S.F.: Multilayer brain networks. *J. Nonlinear Sci.* **30**(5), 2147–2169 (2018). <https://doi.org/10.1007/s00332-017-9436-8>
22. Wagenaar, D., DeMarse, T.B., Potter, S.M.: MeaBench: a toolset for multi-electrode data acquisition and on-line analysis. In: 2005 2nd International IEEE EMBS Conference on Neural Engineering, Conference Proceedings, pp. 518–521 (2005)
23. Yuste, R.: From the neuron doctrine to neural networks. *Nat. Rev. Neurosci.* **16**(8), 487–497 (2015)
24. Zeldenrust, F., Wadman, W.J., Englitz, B.: Neural coding with bursts-current state and future perspectives. *Front. Comput. Neurosci.* **6**(12), 48 (2018)



Complex Networks Reveal Biological Functions of START Domains in Rice: Insights from Computational Systems Biology

Sanjeet Kumar Mahtta , Citu, Abhishek Prasad,
and Gitanjali Yadav

Computational Biology Laboratory, National Institute of Plant Genome Research, New Delhi 110067, India
gy@nipgr.ac.in

Abstract. With the advancement of high throughput technologies, there has been a massive surge in the omics data generation and there is a growing need to integrate this data gain insights into the patterns that shape biological interactions. Complex networks not only enable representation of such interactions, but also offer a mechanism for visualization and understanding of big data at a systems level. In this work, we use a complex network approach to investigate functions of the plant amplified StAR-related lipid transfer (START) domains in rice. We analyse the data at three levels; namely the transcriptome, proteome, and regulome. Each of these distinct datasets was superimposed after generation of the respective co-expression, protein-protein interaction, and gene regulatory networks for rice START genes, to reveal domain specific features in the family. This work thus serves as a protocol for network-based approaches to understand biological processes of genes and to complement experimental strategy.

Keywords: START domain · Rice · Gene co-expression network (GCN) · Protein-Protein interaction network (PPIN) · Gene regulatory network (GRN)

1 Introduction

Plants are fascinating in terms of the incredible diversity of mechanisms they have evolved to cope with changing environment and numerous stresses despite being rooted to the spot. Many of these mechanisms involve changes at the biomolecular level and can be observed and analyzed by studying the genes and their products in terms of DNA, RNA, proteins and metabolites. Advances in high throughput (HTP) sequencing have brought about a new era of big data observation and analyses, where techniques like transcriptomics, metabolomics, proteomics etc. are used to enable capture study the changes in the whole plant system at once (Mochida and Shinozaki 2011). Despite a large amount of available HTP datasets, one of the major issues is that almost 20%–50% of all the genes within genomes are not annotated functionally. Thus, it is of prime importance to assign functions to the large volume of genes that remain unannotated (Mazandu and Mulder 2012). Analysis of high-throughput data generated through RNAseq and microarrays by using computational

methods such as network-based approaches can be effective in the functional annotation of genes in an organism. Systems-level analysis of biological processes became possible after the recent advances in molecular biosciences. Currently, two major approaches are used for systems analysis and building models that have predictive value, one using ‘omics’ data, and the other using pre-defined network properties as well as kinetic properties of the network components.

Biological networks are the starting points for different kinds of analyses made for the understanding of biological systems (Browne et al. 2009). DNA, RNA, proteins, metabolites etc. are the entities that are used for network making. Their structure as well as their interactions can be represented as networks, where the entities become the nodes, and the edges represent the relationship existing between the nodes (Zhang et al. 2014). Cytoscape is an open software that enables the integration of interactive biological networks with expression data generated by high throughput techniques. This software provides three basic functions, which are, providing the network layout, integration of the expression data with the network, and linking the functional annotation databases to the generated network (Shannon et al. 2003). Cytoscape has been used widely for data and interaction analyses of different biological networks, including co-expression networks (GCN), protein-protein interaction (PPI) networks, gene regulatory networks (GRN), metabolic networks etc.

In this study, we have assessed the functions of previously identified START domain-containing proteins (START proteins) and its encoding genes (START genes). The steroidogenic acute regulatory protein (StAR)-related lipid transfer (START) domain is a protein domain of around 200–210 amino acids, known to bind and transport lipid/sterol to the inner mitochondrial membrane in animals (Bork and Koonin 1996; Ponting and Aravind 1999). Although START proteins are abundant in plants relative to animals and have a diverse function in plant processes, knowledge of the role and function of START domains is limited in plants compared to animals, as there are fewer studies in plants. Some Homeodomain associated START genes have been extensively studied in *Arabidopsis*, rice, soybean etc., but the functions of PH (Pleckstrin Homology), DUF (Domain of unknown functions) associated STARTs and minimal START genes are an unexplored area in plants, especially in rice. In plants, studies showed that the five genes namely PHV (*Phavoluta*), PHB (*Phabulosa*), REV (*Revoluta*), ATHB8 and CAN (*Corona*), which belongs to MEKHLA associated HD START (also known as class III HD-ZIP family in *Arabidopsis*) shows multiple and partially overlying roles in development and embryonic patterning of the shoot meristem (Prigge et al. 2005). Similarly, the HD associated START (HD START), which belongs to the class IV HD-ZIP family, shows roles in layer-specific cell differentiation. PDF2 (Protodermal Factor 2) and ATML1 (*Arabidopsis thaliana* Meristem Layer1) show redundant functions in epidermal differentiation in *Arabidopsis* (Lu et al. 1996; Abe et al. 2003). ROC1 (Rice Outer Most Cell-Specific Gene1) of rice has a similar role as ATML1, but its expression is limited to the outermost epidermal layer during early embryogenesis (Ito et al. 2002). OSTF1 (*Oryza sativa* transcription factor 1) is also expressed preferentially in the epidermis and seems to be developmentally regulated during early stage of embryogenesis (Yang et al. 2002). One common expectation is that, since HD START proteins act as a transcription factor in plants, the START domains may regulate the gene expression similar to steroid hormone receptors

from animals when it binds to lipid ligands (sterols). This has been experimentally proven by domain swap experiments in yeast (Schrick et al. 2004; Schrick et al. 2014).

Rice is a global staple food crop and more than half of the world population is dependent on it for dietary requirements. It has now become a necessity to ensure an increase in its production as well as to make it more resistant and tolerant to abiotic and biotic stresses. To fulfill such demands, in-depth genetic and genomic exploration of rice is needed, and thus, a high quality fully sequenced rice genome was imperative. This breakthrough was achieved in 2004, when rice genome was fully sequenced, and this achievement led to a new era of rice functional genomics research (Sasaki 2005). Further, to decipher the regulatory mechanisms leading to increased production and stress resistance, functional characterization of transcription factors is crucial and is currently pursued by many researchers. In this paper, we have described the network-based functional assessment of START genes that were grouped based on co-occurring domains. The first sections explore the identification of co-expressing and interacting partners of START genes and visualization of networks. In later parts of the paper, we have made the gene regulatory networks for the transcription factors that regulate the START genes.

2 Methodology

2.1 Selection and Classification of START Protein-Encoding Genes in Rice

28 START genes for *Oryza sativa var. japonica* (hereafter called ‘rice’) were selected from our previously published work after a comparative analysis of START domains across *Oryza* pangenome (Mahtha et al. 2021). We assigned the genes into seven structural classes based on the presence of co-occurring domains and their spatial arrangements. The comparative domain patterns, gene structure, duplication patterns and expression study helped to streamline these seven structural classes into four major groups, namely (i) MEKHLA associated START genes (HD bZIP START MEKHLA; HZSM) (ii) Only HD associated START genes (HD START; HS and HD bZIP START; HZS) (iii) DUF associated START genes (PH START DUF1336; PSD and START DUF1336; SD) and (iv) minimal START genes; mS (with no known co-occurring domains). Two additional SM (START MEKHLA) genes were identified and each showed strong homology with HZSM and minimal START, and therefore, were clubbed with their respective groups.

2.2 Identification of Putative Co-expressing and Interacting Partners for START Genes

Gene-pair associations and clusters, that may indicate the coordination between biological processes, can be analyzed at a large scale using the co-expression networks. The co-expression network analysis of 28 START genes was carried out for different anatomical parts and development stages. The Genevestigator platform was

used to identify the Co-expressing partners using locus ids of 28 START genes (Hruz et al. 2008) at Pearson correlation coefficient (PCC) value ≥ 0.90 . To understand the major molecular functions and biological processes in which the co-expressing genes are involved, the functional annotation of co-expressing partners were obtained from the Rice genome annotation project (RGAP) (Kawahara et al. 2013) and superimposed on the gene co-expression data. Networks were visualized and analyzed using Cytoscape v 3.8.2 (Shannon et al. 2003).

The protein-protein interaction data was obtained from STRING (Search Tool for the Retrieval of Interacting Genes/Proteins) (Szklarczyk et al. 2019) and PRIN (Predicted Rice Interactome Network) (Gu et al. 2011). PRIN v1 maintains data for 5,049 proteins involved in 76,585 protein-protein interactions reported for *Oryza sativa* (Gu et al. 2011). STRING v11 provides predicted and known associations for more than 5000 organisms (Szklarczyk et al. 2019). The interacting partners for START genes were obtained at high confidence (0.70) score in STRING, while the interacting partners from PRIN were fetched from the supplementary material of the published paper (Gu et al. 2011). To explore the pathways or processes in which these interacting partners are involved, the function of these proteins were retrieved from STRING and literature search and superimposed on the protein-protein interaction data and visualized using Cytoscape v 3.8.2 (Shannon et al. 2003).

2.3 Identification of *cis*-Regulatory Elements and Construction of Gene Regulatory Networks (GRN)

To understand the regulation of START domain encoding genes, a region spanning 1700bp (1500 bp upstream and 200 bp downstream of +1 transcription start site) was extracted using in-house python script. Transcription factor binding sites (TFBS) were identified and putative Transcription Factors (TFs) were predicted on each promoter sequence by MATCHTM tool available with TRANSFAC® professional 12.1 (Matsy 2006). The significantly enriched *cis*-regulatory elements were identified based on the procedure established in our previously published collaborative work on “abiotic stress-tolerant” (ASTR) genes in rice (Smita et al. 2020). To get insights into the gene regulatory networks (GRN) of START genes, the biological processes, molecular function, and KEGG pathways associated with these TFs were identified using ClueGO app (Bindea et al. 2009) in Cytoscape environment (Shannon et al. 2003). Enrichment/depletion (two-headed hyper-geometric test) was then used to generate the networks at p-value to 0.05. Kappa score threshold of 0.4, and Bonferroni adjustment was done for GO terms.

3 Results

3.1 Co-expression Network of START Genes in Rice

Co-expression networks are based on the assumption that the genes that are co-expressed are regulated by the same gene regulatory mechanism.

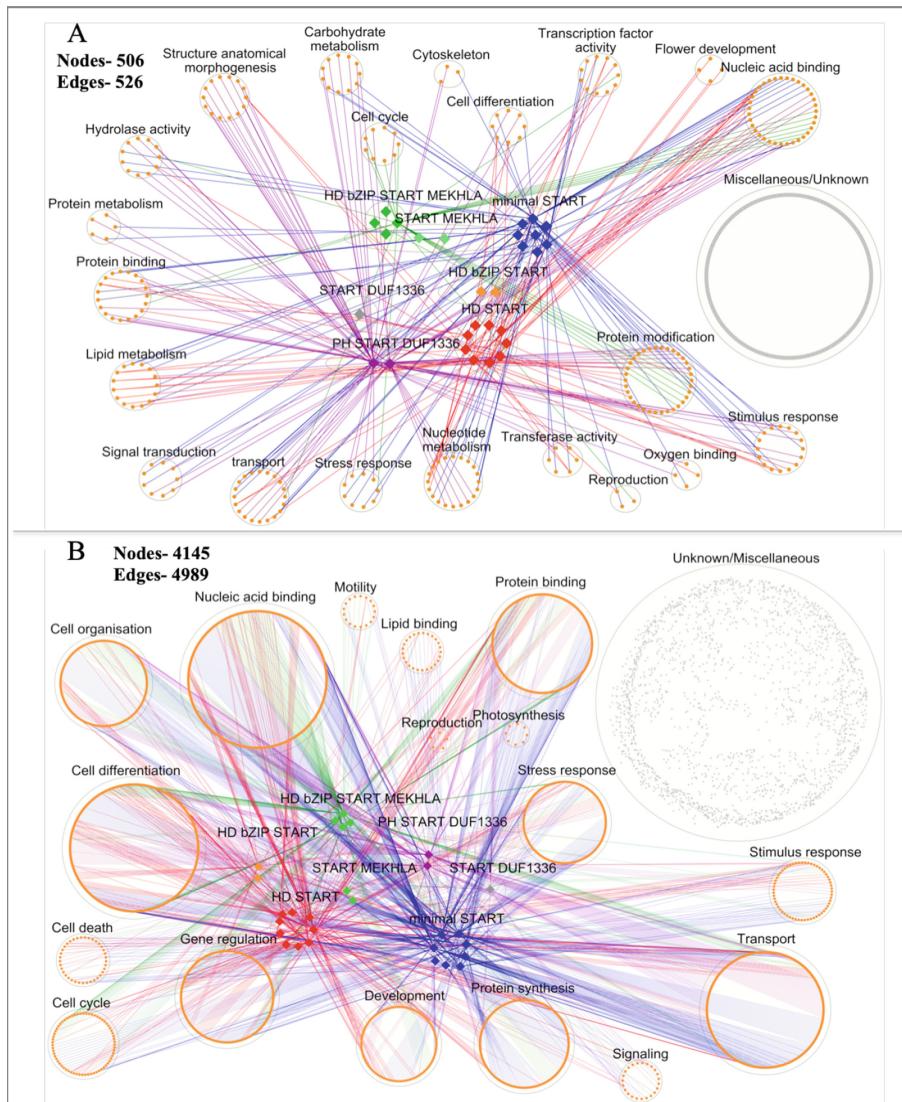


Fig. 1. Gene Co-expression Network of 28 START genes of rice superimposed with respective gene functional annotation. (A) For anatomical parts (B) During developmental stages. The distinct edge colors were used for different groups of START. Green for MEKHLA associated START (HZSM), Red for HD associated START (HS and HZS) purple for DUF associated START (PSD and SD) and blue for minimal STARTs

Thus, analysis of transcriptome at the system level followed by a network analysis can reveal the dynamics among transcripts, indicating their regulation (Weirauch 2011). A total of 514 and 4981 co-expressing partners of 28 START genes were identified for anatomical parts and developmental stages respectively. The functional annotation based gene co-expression network was created and shown in Fig. 1.

As can be seen from Fig. 1, most of the START genes are mainly co-expressed with the genes involved in nucleic acid binding, cell differentiation, transport, gene regulation, protein synthesis and stress response. The classification of 28 START genes into four major structural groups helped in analyzing these networks to find the co-domain class specific patterns. The analysis showed that although all four groups are specifically co-expressed with the genes involved in nucleic acid binding and cell differentiation, some of the class-related patterns are as follows. The HD associated START (HZSM, HZS and HS) were found to be co-expressed with genes mainly enriched in protein modification, lipid and nucleotide metabolism, nucleic acid binding, cell organization, transport and photosynthesis. The DUF associated STARTs (PSD and SD) specifically showed co-expression with genes, which are enriched in processes related to anatomical morphogenesis, cytoskeleton, hydrolase activity, protein binding, and metabolism and stress response. The minimal START co-expressed with the genes involved in wide processes but specifically showed augmentation with photosynthesis, protein synthesis, lipid binding and transport.

3.2 Protein-Protein Interaction Networks of START Genes in Rice

Individual proteins do not function in isolation; they often interact and co-evolve with partners, transiently or in a sustained manner, to achieve their biological functions. Thus, protein-protein interaction networks can be very useful to assign the function of the proteins whose functions remain unknown. This approach has many benefits over other methods as the compilation of information from different sources and then, unifying it into one network increases the coverage of function as well as prediction accuracy (Mazandu and Mulder 2012). To understand the protein interaction networks of START genes, the function-based protein-protein interaction network was generated as described in methods and depicted in Fig. 2. A total of 59 direct interactors were found for 19 START genes through the combined search at PRIN and STRING database. As shown from Fig. 2, the HD associated START (HS and HZS) showed interaction with the proteins which were mainly involved in photosynthesis, protein synthesis, signaling pathways, protein binding, protein degradation and defense signaling pathways or act as dehydration responsive transcription factor. Interestingly, the HD START proteins (HDG12) showed the interaction with three proteins, which are small chain of RuBisCO or a chloroplast precursor. The other group i.e. MEKHLA associated HD START (HZSM) showed interaction with proteins, which were transcriptional factors or cell cycle regulators, seed germination regulators and TFs involved in embryonic developments. Although, in this study, we don't find any interactor for DUF associated PH START domains (PSD), the single interactor for the

START DUF 1336 (SD) showed a role in homeostasis. The single START MEKHLA (SM), which was grouped with HZSM (due to its vast similarity with the latter), showed interactions with the proteins, which act as oxygen sensors, transcription factors, or proteins of the plant cell wall. We also found four interactors of two minimal START, but the functions of these interactors are still unknown.

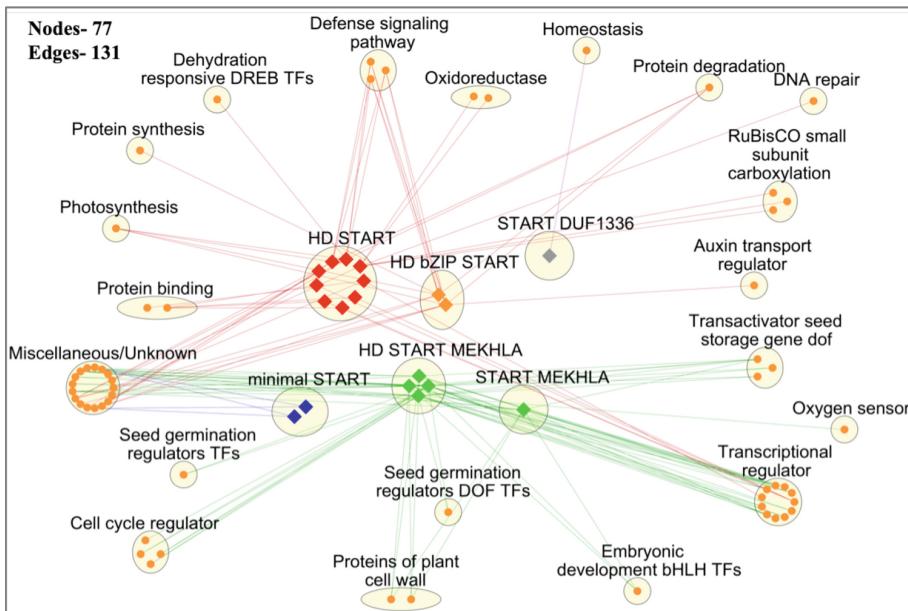


Fig. 2. Protein-protein interaction network (PPIN) of 19 START genes from rice superimposed with respective gene functional annotation. *The edge colors Color codes are same as Fig. 1*

3.3 Gene Regulatory Networks of START Genes in Rice

The term ‘Gene regulatory network’ or GRN, refers to the interconnections between molecular regulators that interact with each other and with other cellular moieties to govern the expression of genes or ultimately functions of mRNA and proteins. GRNs are considered as central to evolutionary developmental biology (evo-devo), in which transcription factors, being the main players in regulatory cascades, serve mainly to activate or regulate genes activities (Peter and Davidson 2011). TFs can turn gene expression on or off by binding to the upstream regions of the Transcription Start Site (TSS), also known as promoter regions/sequences.

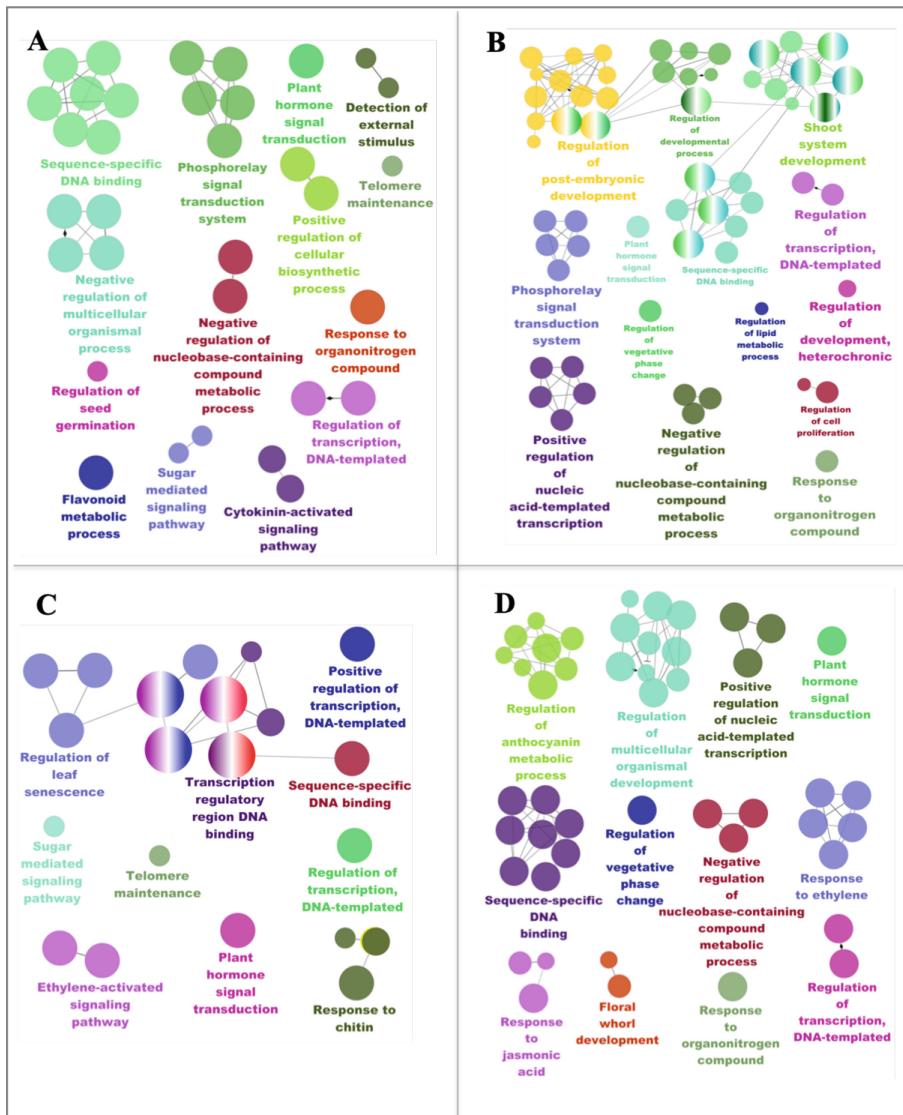


Fig. 3. Gene regulatory networks (GRN) of 28 START genes from rice. (a) MEKHLA associated START; HZSM (b) only HD associated START; HS and HZS (c) DUF associated START; PSD and SD (d) minimal START (mS)

Thus, considering promoters as hubs of regulatory information, we interrogated the upstream cis-regulatory elements of 28 rice START genes, which were clubbed into four major structural groups, as described previously. These cis-elements were analyzed for over-represented biological processes, molecular functions and KEGG pathways through gene set enrichment analysis, as described in methods, and results

are depicted in Fig. 3. As shown in Fig. 3, in addition to the common terms like DNA binding, transcription regulation, sequence-specific DNA binding, etc., all the four classes of START genes showed the association with the biological processes like plant hormone signal transduction, response to organo-nitrogen compound and regulation of cellular biosynthesis. But some of the class-specific functions of MEKHLA associated START domains (HZSM and SM) are in cytokinin activated signaling pathway, detection of external stimulus and flavonoid metabolic process. The HD associated START (HS and HZS) are specifically involved in regulatory processes such as post-embryonic development, cell proliferation and lipid metabolic processes. The DUF associated START (PSD and SD) was found to be specifically associated with ethylene activated signaling pathway, response to chitin and sugar mediated signaling. Interestingly, the minimal START genes also showed diverse enrichment. Most of the specific functions associated with START genes include floral whorl development, regulation of anthocyanin metabolic processes, and response to ethylene and jasmonic acid.

4 Conclusion and Future Perspective

Complex networks offer a systems-level understanding of biological interactions, while making it convenient to visualize unipartite as well as multi-partite functional relationships among proteins, gene, metabolites etc. These are especially useful for analysis of biological processes that are often dynamic, changing under different conditions as well as in different physiological states. Each of these changes modifies the topology and structure of the network, and these changes can be quantified using node, edge and network parameters, further aiding functional interpretations. In this work, we have assessed domain specific functions of START gene family in rice using biological networks. The 28 genes in this family were classified into four major groups based on the co-occurrence of domains and three distinct networks were generated based on co-expression, protein interaction and gene regulatory data. Superimposition of functional data on these networks revealed interesting features, enabling assessment of domain-specific functions through protein interactions and gene regulatory networks. This work can be used as a protocol paper for network-based approaches to understand biological processes of genes and to complement experimental work pipelines.

Acknowledgement. The authors acknowledge support of National Institute of Plant Genome Research (NIPGR), New Delhi for funding and infrastructure.

Funding. SKM and C acknowledge the Department of Biotechnology (DBT) Government of India for PhD fellowship. AP received fellowship from the CSIR. Participation in CNA was funded by RCUK BBSRC Grant BB/P027970/1TIGR2ESS ‘Transforming India’s Green Revolution by Research and Empowerment for Sustainable food Supplies’.

References

- Abe, M., Katsumata, H., Komeda, Y., Takahashi, T.: Regulation of shoot epidermal cell differentiation by a pair of homeodomain proteins in *Arabidopsis*. *Development* **130**, 635–643 (2003). <https://doi.org/10.1242/dev.00292>
- Bindea, G., et al.: ClueGO: a cytoscape plug-in to decipher functionally grouped gene ontology and pathway annotation networks. *Bioinformatics* **25**, 1091–1093 (2009). <https://doi.org/10.1093/bioinformatics/btp101>
- Bork, P., Koonin, E.V.: Protein sequence motifs. *Curr. Opin. Struct. Biol.* **6**, 366–376 (1996). [https://doi.org/10.1016/S0959-440X\(96\)80057-1](https://doi.org/10.1016/S0959-440X(96)80057-1)
- Browne, F., Wang, H., Zheng, H., Azuaje, F.: GRIP: a web-based system for constructing gold standard datasets for protein-protein interaction prediction. *Source Code Biol. Med.* **4**, 2 (2009). <https://doi.org/10.1186/1751-0473-4-2>
- Gu, H., Zhu, P., Jiao, Y., Meng, Y., Chen, M.: PRIN: a predicted rice interactome network. *BMC Bioinforma* **12**(12), 1–13 (2011). <https://doi.org/10.1186/1471-2105-12-161>
- Hruz, T., et al.: Genevestigator V3: a reference expression database for the meta-analysis of transcriptomes. *Adv. Bioinform.* **2008**, 5 (2008). <https://doi.org/10.1155/2008/420747>
- Ito, M., Sentoku, N., Nishimura, A., Hong, S.K., Sato, Y., Matsuoka, M.: Position dependent expression of gl2-type homeobox gene, roc1: Significance for protoderm differentiation and radial pattern formation in early rice embryogenesis. *Plant J.* **29**, 497–507 (2002). <https://doi.org/10.1046/j.1365-313x.2002.01234.x>
- Kawahara, Y., et al.: Improvement of the *Oryza sativa* Nipponbare reference genome using next generation sequence and optical map data. *Rice* **6**(1), 1–10 (2013). <https://doi.org/10.1186/1939-8433-6-4>
- Lu, P., Porat, R., Nadeau, J.A., O'Neill, S.D.: Identification of a meristem L1 layer-specific gene in *Arabidopsis* that is expressed during embryonic pattern formation and defines a new class of homeobox genes. *Plant Cell* **8**, 2155–2168 (1996). <https://doi.org/10.1105/tpc.8.12.2155>
- Mahtha, S.K., Purama, R.K., Yadav, G.: StAR-Related Lipid Transfer (START) domains across the rice Pangenome reveal how ontogeny recapitulated selection pressures during rice domestication. *Front. Genet.* **12**, 1658 (2021). <https://doi.org/10.3389/fgene.2021.737194>
- Matys, V.: TRANSFAC(R) and its module TRANSCompel(R): transcriptional gene regulation in eukaryotes. *Nucl. Acids Res.* **34**, D108–D110 (2006). <https://doi.org/10.1093/nar/gkj143>
- Mazandu, G.K., Mulder, N.J.: A topology-based metric for measuring term similarity in the gene ontology. *Adv. Bioinform.* **2012**, 783 (2012). <https://doi.org/10.1155/2012/975783>
- Mochida, K., Shinozaki, K.: Advances in omics and bioinformatics tools for systems analyses of plant functions. *Plant Cell Physiol.* **52**, 2017–2038 (2011). <https://doi.org/10.1093/PCP/PCR153>
- Peter, I.S., Davidson, E.H.: Evolution of gene regulatory networks that control embryonic development of the body plan. *Cell* **144**, 970 (2011). <https://doi.org/10.1016/J.CELL.2011.02.017>
- Ponting, C.P., Aravind, L.: START: a lipid-binding domain in StAR, HD-ZIP and signalling proteins. *Trends Biochem. Sci.* **24**, 130–132 (1999). [https://doi.org/10.1016/S0968-0004\(99\)01362-6](https://doi.org/10.1016/S0968-0004(99)01362-6)
- Prigge, M.J., Otsuga, D., Alonso, J.M., Ecker, J.R., Drews, G.N., Clark, S.E.: Class III homeodomain-leucine zipper gene family members have overlapping, antagonistic, and distinct roles in *Arabidopsis* development. *Plant Cell* **17**, 61–76 (2005). <https://doi.org/10.1105/tpc.104.026161.1>
- Sasaki, T.: The map-based sequence of the rice genome. *Nature* **436**(7052), 793–800 (2005). <https://doi.org/10.1038/nature03895>

- Schrick, K., et al.: Shared functions of plant and mammalian StAR-related lipid transfer (START) domains in modulating transcription factor activity. *BMC Biol.* **12**(1), 1–20 (2014). <https://doi.org/10.1186/s12915-014-0070-8>
- Schrick, K., Nguyen, D., Karlowski, W.M., Mayer, K.F.X.: START lipid/sterol-binding domains are amplified in plants and are predominantly associated with homeodomain transcription factors. *Genom. Biol.* **5**, R41 (2004). <https://doi.org/10.1186/gb-2004-5-6-r41>
- Shannon, P., et al.: Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genom. Res.* **13**, 2498 (2003). <https://doi.org/10.1101/GR.1239303>
- Smita, S., et al.: Gene network modules associated with abiotic stress response in tolerant rice genotypes identified by transcriptome meta-analysis. *Funct. Integr. Genomics* **20**, 29–49 (2020). <https://doi.org/10.1007/s10142-019-00697-w>
- Szklarczyk, D., et al.: STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucl. Acids Res.* **47**, D607 (2019). <https://doi.org/10.1093/NAR/GKY1131>
- Weirauch, M.T.: Gene coexpression networks for the analysis of DNA microarray data. *Appl. Stat. Netw. Biol. Methods Syst. Biol.* **1**, 215–250 (2011). <https://doi.org/10.1002/9783527638079.CH11>
- Yang, J.Y., Chung, M.C., Tu, C.Y., Leu, W.M.: OSTF1: A HD-GL2 family homeobox gene is developmentally regulated during early embryogenesis in rice. *Plant Cell Physiol.* **43**, 628–638 (2002). <https://doi.org/10.1093/pcp/pcf076>
- Zhang, B., Tian, Y., Zhang, Z.: Network Biology in Medicine and Beyond. *Circ. Cardiovasc. Genet.* **7**, 536–547 (2014). <https://doi.org/10.1161/CIRCGENETICS.113.000123>



Supervised Gene Function Prediction Using Spectral Clustering on Gene Co-expression Networks

Miguel Romero^(✉), Óscar Ramírez, Jorge Finke, and Camilo Rocha

Department of Electronics and Computer Science, Pontificia Universidad Javeriana,
Cali, Colombia

miguel.romero@javerianacali.edu.co

Abstract. Gene annotation addresses the problem of predicting unknown functions that are associated to the genes of a specific organism (e.g., biological processes). Despite recent advances, the cost and time demanded by annotation procedures that rely largely on *in vivo* biological experiments remain prohibitively high. This paper presents an *in silico* approach to the annotation of genes that follows a network-based representation, and combines techniques from multivariate statistics (spectral clustering) and machine learning (gradient boosting). Spectral clustering is used to enrich the gene co-expression network (GCN) with currently known gene annotations. Gradient boosting is trained on features of the GCN to build an estimator of the probability that a gene is involved in a given biological process. The proposed approach is applied to a case study on *Zea mays*, one of the world's most dominant and productive crop. Broadly speaking, the main results illustrate how computational experimentation narrows down the time and costs in efforts to annotate the functions of genes. More specifically, the results highlight the importance of network science, multivariate statistics, and machine learning techniques in reducing types I and II prediction errors.

1 Introduction

An important pillar for gaining insight into how genomes serve as blueprints for life is understanding the association of genes with as yet unknown functions [25, 29]. Developing treatments that use genomic information about organisms to treat specific conditions, including—for example—approaches that enhance tolerance levels to environmental stresses, has motivated a significant body of research [28]. Nonetheless, the cost and time demanded by *in vivo* biological experimentation to annotate large sets of genes remains prohibitively high [5, 32]. Hybrid approaches that integrate existing knowledge of gene-function associations and *in silico* methods have been introduced to overcome this limitation [6, 8, 15, 24]. The ability to cope with the extreme combinatorial nature of gene annotation enables computational experimentation to narrow down the effort, time, and costs.

A number of studies have shown that the representation and analysis of gene co-expression networks (GCNs) are a useful framework for guiding *in silico* annotation of genes [20, 27]. The crux of GCN-based analysis is that it exploits the topology of the network and offers a rich source of new information for predicting gene-function

associations [26]. In practice, such approaches remain valid as long as the data for gene–function associations is relatively complete and the underlying co-expression network is tractable. Therefore, unlocking the full potential of network-based gene annotation demands efficient approaches that are scalable to long datasets.

Main Contribution. This paper presents a novel approach for *in silico* annotation of genes. It follows a network-based approximation that uses clustering and machine learning for building a predictor that assigns functions to genes. The role of clustering is to enrich the available information for gene–function associations by creating new features that are later used for supervised learning. More precisely, new features are built by taking into account clusters that seem relevant to the specific biological function under scrutiny. These new features are filtered based on the impact that is made on prediction, before a machine learning algorithm is used to build the predictor. The proposed approach illustrates how the performance of gene annotation is improved based on the new information obtained from the clustering of the GCN.

The approach is applied to a case study on *Zea mays*, the world’s most dominant and productive crop. *Zea mays* is used for a variety of purposes, including animal feed and derivatives for human consumption, and ethanol [31]. The co-expression information used in the case study is borrowed from the ATTED-II database [18]. The resulting GCN, modeled as a weighted graph, comprises 26,131 vertices (genes) and 44,621,533 edges (binary co-expression gene relations). The functional information (known gene–function associations) is taken from AmiGO [2]; it contains annotations of biological processes, i.e., pathways to which a gene contributes. A total of 5,361 genes are associated to 3,285 functions. Two benchmarks are introduced and used to contrast the results of the approach. The comparison highlights the importance of multivariate statistics and machine learning techniques in reducing type I (i.e., false positives) and type II (i.e., false negatives) prediction errors. Ultimately, this case study provides experimental (*in silico*) evidence that the proposed approach is a viable and promising approximation to gene function prediction.

Related Work. The authors in [31] predict functions of maize proteins using graph convolutional networks. In particular, amino acid sequence of proteins and the Gene Ontology (GO) hierarchy are used to predict functions of proteins with a deep graph convolutional network model (DeepGOA). The results show that DeepGOA integrates amino acid data and the GO structure to accurately annotate proteins. The work in [7] aims to predict the phenotypes and functions associated to maize genes using (i) hierarchical clustering based on datasets of transcriptome (set of molecules produced in transcription) and metabolome (set of metabolites found within an organism); and (ii) GO enrichment analyses. The results show that profiling individual plants is a promising experimental design for narrowing down the lab-field gap. Finally, a prediction of protein functions for *Zea mays* is presented in [17]. The approach, called PiZeam, is built using a method of interacting orthologs (genes are said to be *orthologs* if they evolved from a common ancestor). PiZeam demonstrates that the protein functions of maize can be predicted based on protein sequence data of other organisms because orthologs tend to retain the same function [10].

Outline. The remainder of the paper is organized as follows. Section 2 gathers some preliminaries. The proposed approach is presented in Sect. 3. Section 4 presents a case

study for the *Zea mays* species. Finally, Sect. 5 draws some concluding remarks and future research directions.

2 Preliminaries

This section presents preliminaries on gene co-expression networks, gene function prediction, spectral clustering, and the SHAP technique.

2.1 Gene Co-expression Network

A gene co-expression network (GCN) is represented as an undirected graph where each vertex represents a gene and each edge the level of co-expression between two genes.

Definition 1. *Let V be a set of genes, E a set of edges that connect pairs of genes, and $w : E \rightarrow \mathbb{R}_{\geq 0}$ a weight function. A (weighted) gene co-expression network is a weighted graph $G = (V, E, w)$.*

The set of genes V in a co-expression network is particular to the genome under study. The correlation of expression profiles between each pair of genes is measured, commonly, with the help of the Pearson correlation coefficient. Every pair of genes is assigned and ranked according to a relationship measure, and a threshold is used as a cut-off value to determine E . The weight function w denotes how strongly co-expressed are each pair of genes in V . For example, in the ATTED-II database, the co-expression relation between any pair of genes is measured as a z-score expressed as a function of the co-expression index LS (Logit Score) [18, 19].

2.2 Gene Function Prediction

In an annotated gene co-expression network, each gene is associated with the collection of biological functions to which it is related (e.g., through in vivo experiments).

Definition 2. *Let A be a set of biological functions. An annotated gene co-expression network is a gene co-expression network $G = (V, E, w)$ complemented with an annotation function $\phi : V \rightarrow 2^A$.*

The problem of predicting gene functions can be explained as follows. Given an annotated co-expression network $G = (V, E, w)$ with annotation function ϕ , the goal is to use the information represented by ϕ , together with additional information (e.g., features of G), to obtain a function $\psi : V \rightarrow 2^A$ that extends ϕ . Associations between genes and functions not present in ϕ have either not been found through in vivo experiments or do not exist in a biological sense. The new associations identified by ψ are a suggestion of functions that need to be verified through in vivo experiments. The function ψ can be built from a predictor of gene functions, e.g., based on a supervised machine learning model.

2.3 Spectral Clustering

The goal of clustering classification on a network is to identify groups of vertices sharing a (parametric) notion of similarity [23]. Usually, distance or centrality metrics are used for clustering. Spectral clustering is an important clustering method due to its precise foundation from algebraic graph theory [11]. It has been shown that spectral clustering has better overall performance, but with somewhat more instability compared to other algorithms [16]. Given a graph G , the spectral clustering decomposition of G can be represented by the equation $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where \mathbf{L} is the Laplacian, \mathbf{D} is the degree (i.e., a diagonal matrix with the number of edges incident to each node), and \mathbf{A} the adjacency matrices of G . This technique uses, say, the n eigenvectors associated to the n smallest nonzero eigenvalues of \mathbf{L} . In this way, each node of the graph gets a coordinate in \mathbb{R}^n . The resulting collection of eigenvectors serve as input to a clustering algorithm (e.g., k-means) that groups the nodes in n clusters.

2.4 SHAP

In general, the performance of classification algorithms is determined by the features used for training a particular prediction. SHAP (SHapley Additive exPlanation) is a framework that allows us to compute importance values for each feature using concepts from game theory [13]. Given a predictor and a training set, SHAP assigns Shapely values to explain which features in the model are the most important for prediction by calculating the changes in the prediction when features are conditioned. A key advantage of SHAP is that its plots depict the contributions of different weights of features in a predictor [14].

3 Clustering-Based Function Prediction

This section presents the approach for gene function prediction based on spectral clustering. The approach combines multivariate statistics and supervised learning techniques to create a predictor enriched with the information of clusters. The predictor takes into account features that capture topological properties of the GCN.

The approach can be independently applied to each function in the set of gene functions A to be predicted. Formally, the inputs of the approach are a GCN, denoted by $G = (V, E, w)$, an annotation function $\phi : V \rightarrow 2^A$, a (biological) function $a \in A$, a set $K = \{k_0, \dots, k_{m-1}\}$ for sampling the number of clusters, and a constant value $c \in [0, 1]$ for feature selection. The output is a function $\psi_a : V \rightarrow [0, 1]$, which indicates for each gene $v \in V$, the probability $\psi_a(v)$ of v having the function a .

The proposed approach consists of four stages. First, an enriched graph with information in ϕ is created from G . Second, m features are created for both G and its enriched version obtained in the previous stage, corresponding to the m number of clusters in K . Third, these new features are filtered by selecting those with more impact in the prediction task. Fourth, supervised learning is used to build the predictor ψ_a . These stages are depicted and detailed in Fig. 1. They have been implemented in Python and are available at <https://github.com/migueleci/geneclust>. The rest of this section is devoted to detailing each of the four stages in the approach.

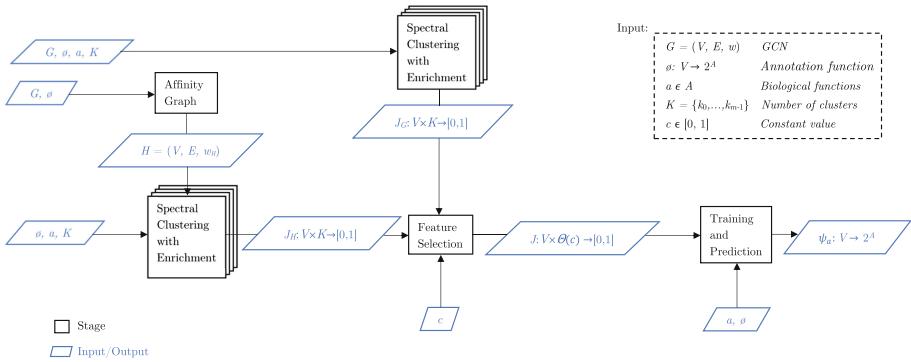


Fig. 1. The clustering-based approach is split into four stages. Namely, creation of affinity graph, clustering computation, feature selection, and training and prediction. Its inputs are a GCN, denoted by $G = (V, E, w)$, an annotation function $\phi : V \rightarrow 2^A$, a function $a \in A$, a set $K = \{k_0, \dots, k_{m-1}\}$, and a real number $c \in [0, 1]$. Its output is a predictor ψ_a , which indicates, for each gene $v \in V$, the probability $\psi_a(v)$ of v having function a .

3.1 Affinity Graph Creation

An affinity graph $H = (V, E, w_H)$ between G and ϕ is built. Its weight function is defined as the mean between the co-expression weight specified by w and the proportion of shared functions between genes as specified by ϕ .

Definition 3. *The weight function $w_H : V \times V \rightarrow [0, 1]$ is defined for any $u, v \in V$ as*

$$w_H(u, v) = \frac{1}{2} \left(\frac{w(u, v) - 1}{\max(w) - 1} + \frac{|\phi(u) \cup \phi(v)|}{|\phi(u) \cap \phi(v)|} \right),$$

where $\max(w)$ denotes the maximum value in the range of w (which exists because w is finite).

It is guaranteed that the range of w_H is $[0, 1]$ under the assumption that at least one element in the range of w is greater than 1 because $w : V \times V \rightarrow [1, \infty)$. This is indeed the case, in practice, because the co-expression between two genes in the GCN is quantified in terms of the z -score, which is highly unlikely to be 1 for all pairs of genes.

3.2 Gene Enrichment with Clustering

For each graph $X \in \{G, H\}$, the goal of this stage is to produce a matrix $J_X : V \times K \rightarrow [0, 1]$ that specifies how likely it is for the genes to be associated to function a when X is decomposed in a given number of clusters. This is achieved in two steps, namely, by using clustering and computing a measure for each node in each cluster.

The decomposition of X is performed m times, once for each value k in K . The approach uses spectral clustering for finding the k clusters. The output of the clustering

algorithm is an assignment from nodes to clusters. Each cluster is used to gather and compute information with the goal of deciding whether a significant number of members associated to function a is (locally) present. Intuitively, if genes grouped together have a strong co-expression relation and most of them in the group are associated to gene function a , the remaining genes are also likely to be associated to a (guilt by association, see [21]). In this way, for each $v \in V$ and $k \in K$, the entry $J_X(v, k)$ specifies a p -value indicating if the function a is over-represented in the decomposition of k clusters of X . This process is commonly known as Gene Ontology Term Enrichment and may use different statistical tests, such as, Fisher's exact test [30].

3.3 Feature Selection

Matrices J_G and J_H represent structural properties of the GCN. They also represent associations between genes and functions based on partitions of each graph. The goal of this stage is to produce a matrix $J : V \times \Theta(c) \rightarrow [0, 1]$ by selecting a reduced number of significant features $0 \leq \Theta(c) \leq 2m$ from J_G and J_H .

Feature selection is conveyed from J_G and J_H to J using SHAP. Let J_{G+H} denote the matrix resulting from extending J_G with the m columns of J_H . That is, for each $v \in V$, the expression $J_{G+H}(v, \cdot)$ denotes a function with domain $[0, 2m)$ and range $[0, 1]$, where the values in $[0, m)$ denote the p -values associated to v in G and the values in $[m, 2m)$ the ones associated to v in H . For each entry $J_{G+H}(v, j)$, with $v \in V$ and $0 \leq j < 2m$, the mean absolute SHAP value $s_{(v,j)}$ is computed after a large enough number of Shapely values are computed for feature j (executions of SHAP). Features are selected based on the cutoff

$$c \cdot \sum_{j=0}^{2m-1} s_{(v,j)},$$

i.e., on the sum of mean absolute values by a factor of the input constant c . The first $\Theta(c)$ features, sorted from greater to lower mean absolute SHAP value, are selected as to reach the given cutoff.

Note that the input constant c is key for selecting the number of significant features. The idea is to set c so as to find a balance between prediction efficiency and the computational cost of building the predictor.

3.4 Training and Prediction

This stage comprises a process that combines different supervised machine learning techniques/tools to finally build the predictor ψ_a . In particular, stratified k -fold cross-validation, the Synthetic Minority Over-sampling Technique (SMOTE) [3], hyper-parameter tuning [1], and XGBoost [4] are used sequentially in a pipeline. Stratified k -fold and over-sampling aim to overcome overfitting and learning bias. SMOTE is used to handle imbalanced datasets for underrepresented classes; it synthesizes new examples of the minority class from the existing ones. Hyper-parameter tuning aims to improve the performance of the prediction by optimizing parameters of the classifier such as, e.g., learning rate and maximum depth of trees. Gradient boosting decision tree algorithm, namely, XGBoost [4], is used as classifier.

The pipeline takes as input the matrix J , which specifies the significant features of J_G and J_H , the annotation function ϕ , and the set $\phi^{-1}(a)$ of genes associated to a . First, k -fold is applied to split the dataset into k different folds (this k has nothing to do with the input K). Each fold is used as test set, while the remaining $k - 1$ folds are used for training. Furthermore, the training set is balanced using SMOTE to over-sample the minority class. The balanced training set is used to tune the following hyper-parameters of the XGBoost classifier: maximum tree depth for base learners, minimum sum of instance weight needed in a child, boosting learning rate, and subsample ratio of the training instance.

The prediction is carried out using the best estimator, i.e., the estimator with the combinations of parameters' values that achieve the best performance. The output are the probabilities of associations between the genes in V and function a , namely, the predictor ψ_a .

4 Case Study: *Zea Mays*

Next section describes a case study on applying the approach presented in Sect. 3 to maize (*Zea mays*). First, the maize data used for the study is described. Second, two benchmarks are introduced to compare the performance of the approach. Finally, the outcome of the proposed approach is contrasted with the benchmarks.

4.1 Data Description and Feature Selection

The co-expression information used in the study is borrowed from the ATTED-II database [18]. The gene co-expression network $G = (V, E, w)$ comprises 26,131 vertices (genes) and 44,621,533 edges. In this case, a z -score threshold of 1 is used as the cut-off measure for G , i.e., E contains edges e that satisfy $w(e) \geq 1$ (most of them satisfying $w(e) > 1$). Note that the highest value is assigned to the strongest connections. The functional information for this network is taken from AmiGO [2]; it contains annotations of biological processes, i.e., pathways to which a gene contributes. It is important to note that genes may be associated to several biological processes. A total of 5,361 genes are associated to 3,285 functions, comprising 20.5% of the genes in V .

Only 121 (3.7%) functions are associated to more than 10% of the genes in the GCN; that is, the data is highly imbalanced in most cases. For this reason, only biological processes corresponding to functions of level 1 in the function hierarchy defined by [9] are used for the prediction. For example, if a gene is associated to the function *response to light stimulus* and *response to stimulus* is its ancestor of level 1, then the gene will be associated to *response to stimulus*. Note that the functions of level 1 in the hierarchy are the more general ones.

As result, there are 19 biological processes of level 1, twelve of which are associated to more than 40 genes. Thus, the final dataset of associations between genes and functions is more balanced for applying supervised learning. Table 1 lists the 12 biological processes used for the prediction. In the end, the function $\phi : V \rightarrow 2^A$ for G associates $|A| = 12$ functions (biological processes of level 1 in function hierarchy) to 5,361 genes. The remaining input parameters are $K = \{10, 20, \dots, 100\}$ and $c = 0.9$.

Table 1. Biological processes A of level 1 in the Gene Ontology hierarchy [9] used for prediction. The identifier and name of each function is presented in the first and second columns, respectively. The third column shows the number of maize genes associated to each function.

Term	Description	Genes	% of GCN
GO:0009987	Cellular process	4,269	16.34
GO:0008152	Metabolic process	3,047	11.66
GO:0065007	Biological regulation	1,492	5.71
GO:0051179	Localization	872	3.34
GO:0050896	Response to stimulus	851	3.26
GO:0023052	Signaling	307	1.17
GO:0032502	Developmental process	124	0.47
GO:0000003	Reproduction	78	0.30
GO:0032501	Multicellular organismal process	76	0.29
GO:0022414	Reproductive process	76	0.29
GO:0044419	Biological process involved in interspecies interaction between organisms	48	0.18
GO:0002376	Immune system process	47	0.18

Figure 2 depicts the distribution of filtered features using SHAP for each function in A . Note that, in most cases, the features coming from the affinity graph H are more important (or have more impact) for the prediction task.

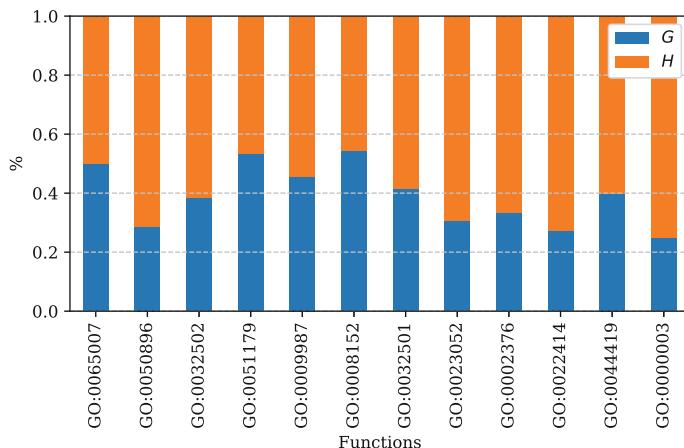


Fig. 2. Distribution of selected features using SHAP, from the 20 features corresponding to $K = \{10, 20, \dots, 100\}$ for G and H . The features from the affinity graph H have more impact than those from G .

4.2 Benchmarks

Two models are built to benchmark the performance of the proposed approach. First, a baseline model using only the matrix of associations between genes and functions (i.e., matrix representation of function ϕ) as input features for the prediction. Second, a model including some topological properties of the GCN as additional features (together with the representation of ϕ) to train the predictor.

The topological properties included for each gene in V are the degree, average neighbor degree, eccentricity, clustering coefficient, closeness centrality, betweenness centrality, PageRank, Kleinberg's authority score, Kleinberg's hub scores, and coreness. These measures were computed with the help of igraph [12], an open source and free collection of network analysis tools available in several programming languages.

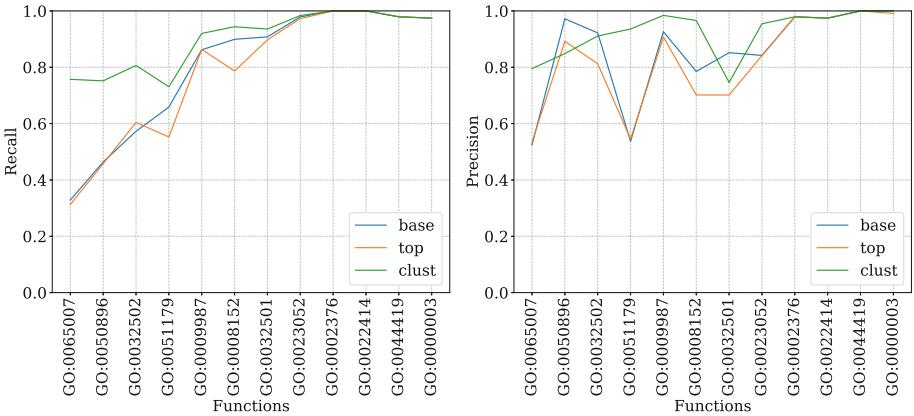


Fig. 3. Prediction performance measured with recall and precision score for the prediction of the 12 functions in A with the proposed approach and the benchmarks. The approach is labeled as *clust*, the baseline benchmark is labeled as *base*, and the one including the topological properties is labeled as *top*.

4.3 Summary of Results

Figure 3 presents the prediction performance of the proposed approach and its comparison to the benchmarks by using the recall and precision scores. It can be seen that the proposed approach outperforms both benchmarks in terms of the recall score. That is, the proposed approach is better at identifying the associations between genes and functions even though datasets are highly imbalanced in some cases. For example, recall scores for the function GO:0065007 are 0.76, 0.33, and 0.31 for *clust*, *base*, and *top*, respectively. That is, the proposed approach improves the performance in relation to the benchmarks.

The precision score measures how many of the predicted associations are relevant (i.e., true w.r.t. ϕ). Therefore, predictions that are not part of ϕ are considered as suggestions of possible gene-function associations. For this reason, the recall score is relevant

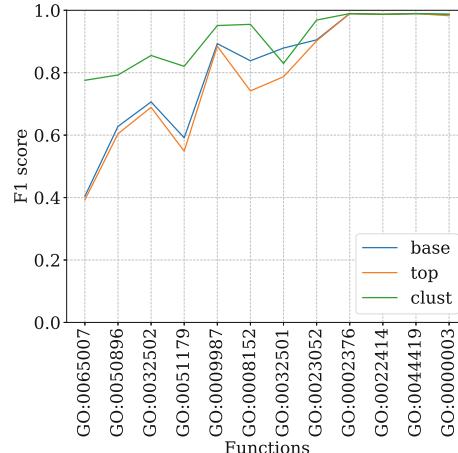


Fig. 4. Prediction performance measured by the F1 score with the proposed approach *clust*, and the benchmarks *base* and *top*.

to measure the performance of the predictions in relation to the known associations in ϕ . Note that both (recall and precision) scores of the proposed approach are higher than 0.75 for all functions A. Furthermore, the precision score of the proposed approach is lower than the benchmarks for 3 functions. In these cases, false positives can be considered to suggest candidate associations.

Figure 4 presents the F1 scores for the proposed approach and the benchmarks. Recall that the F1 score is the harmonic mean of precision p and recall r , and it is defined as $F1 = \frac{2pr}{p+r}$. The F1 score of the proposed approach is at least as good as the base benchmark for 11 out 12 functions. The exception is in relation to the GO:0032502 function.

As final word on the choice of the clustering algorithm, it must be noted that other clustering algorithms, such as DBSCAN [22], were evaluated. However, spectral clustering showed the best and more consistent performance.

5 Concluding Remarks and Future Work

By combining network-based modeling, clustering, and supervised machine learning, the approach presented in this paper introduces a novel method to address the gene function prediction problem. It aims to predict the association probability between each gene and function, taking advantage of the GCN spectral decomposition, and the information available of associations between genes and functions. A comparison between the proposed approach and two benchmarks on a *Zea mays* case study is presented. Using the structural information of the network, computed by a spectral clustering algorithm, is likely to be the key for the good performance of other GCN-based predictors. The proposed approach outperforms the two benchmarks, especially in terms of the recall score.

Two main lines of work can be considered for future work. First, gathering more information of associations between genes and functions for *Zea mays* is required. This way, it would be possible to use the functions beyond level 1 in the Gene Ontology hierarchy, therefore including more specific functions and their corresponding hierarchical constraints. Second, applying the proposed approach to identify genes associated to specific stresses, such as low temperature, can help to reduce the set of candidate genes that respond to treatments for in vivo validation.

Acknowledgments. This work was partially funded by the OMICAS program: Optimización Multiescala In-silico de Cultivos Agrícolas Sostenibles (Infraestructura y Validación en Arroz y Caña de Azúcar), anchored at the Pontificia Universidad Javeriana in Cali and funded within the Colombian Scientific Ecosystem by The World Bank, the Colombian Ministry of Science, Technology and Innovation, the Colombian Ministry of Education and the Colombian Ministry of Industry and Turism, and ICETEX, under GRANT ID: FP44842-217-2018. The second author was partially supported by Fundación CeiBA.

References

1. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **13**(10), 281–305 (2012)
2. Carbon, S., Mungall, C.: Gene Ontology Data Archive, July 2018. Type: dataset
3. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002)
4. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
5. Cho, H., Berger, B., Peng, J.: Diffusion component analysis: unraveling functional topology in biological networks. In: Przytycka, T.M. (ed.) RECOMB 2015. LNCS, vol. 9029, pp. 62–64. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16706-0_9
6. Cho, H., Berger, B., Peng, J.: Compact integration of multi-network topology for functional analysis of genes. *Cell Syst.* **3**(6), 540–548.e5 (2016)
7. Cruz, D.F., et al.: Using single-plant-omics in the field to link maize genes to functions and phenotypes. *Mol. Syst. Biol.* **16**(12), e9667 (2020)
8. Deng, M., Zhang, K., Mehta, S., Chen, T., Sun, F.: Prediction of protein function using protein-protein interaction data. *J. Comput. Biol.* **10**(6), 947–960 (2003)
9. Gene Ontology Consortium: The gene ontology resource: 20 years and still GOing strong. *Nucl. Acids Res.* **47**(D1), D330–D338 (2019)
10. Jensen, R.A.: Orthologs and paralogs - we need to get it right. *Genom. Biol.* **2**(8), 1–3 (2001)
11. Jia, H., Ding, S., Xu, X., Nie, R.: The latest research progress on spectral clustering. *Neural Comput. Appl.* **24**(7), 1477–1486 (2013). <https://doi.org/10.1007/s00521-013-1439-2>
12. Ju, W., Li, J., Yu, W., Zhang, R.: iGraph: an incremental data processing system for dynamic graph. *Front. Comput. Sci.* **10**(3), 462–476 (2016)
13. Lundberg, S., Lee, S.-I.: A Unified Approach to Interpreting Model Predictions. [arXiv:1705.07874](https://arxiv.org/abs/1705.07874), November 2017
14. Lundberg, S.M., et al.: From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2**(1), 2522–5839 (2020)
15. Luo, F., et al.: Constructing gene co-expression networks and predicting functions of unknown genes by random matrix theory. *BMC Bioinform.* **8**(1), 299 (2007)

16. Murugesan, N., Cho, I., Tortora, C.: Benchmarking in cluster analysis: a study on spectral clustering, DBSCAN, and K-Means. In: Chadjipadelis, T., Lausen, B., Markos, A., Lee, T.R., Montanari, A., Nugent, R. (eds.) IFCS 2019. SCDAKO, pp. 175–185. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-60104-1_20
17. Musungu, B., Bhatnagar, D., Brown, R.L., Fakhouri, A.M., Geisler, M.: A predicted protein interactome identifies conserved global networks and disease resistance subnetworks in maize. *Front. Genet.* **6** (2015)
18. Obayashi, T., Aoki, Y., Tadaka, S., Kagaya, Y., Kinoshita, K.: ATTED-II in 2018: a plant coexpression database based on investigation of the statistical property of the mutual rank index. *Plant Cell Physiol.* **59**(1), e3–e3 (2018)
19. Obayashi, T., Kinoshita, K.: COXPRESdb: a database to compare gene coexpression in seven model animals. *Nucl. Acids Res.* **39**, D1016–D1022 (2011)
20. Oti, M., van Reeuwijk, J., Huynen, M.A., Brunner, H.G.: Conserved co-expression for candidate disease gene prioritization. *BMC Bioinform.* **9**(1), 208 (2008)
21. Petsko, G.A.: Guilt by association. *Genom. Biol.* **10**(4), 104 (2009)
22. Rehman, S.U., Asghar, S., Fong, S., Sarasvady, S.: DBSCAN: past, present and future. In: The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014), pp. 232–238, Bangalore, India, February 2014
23. Rodriguez, M.Z., et al.: Clustering algorithms: a comparative approach. *PLoS One* **14**(1), e0210236 (2019)
24. Romero, M., Finke, J., Quimbaya, M., Rocha, C.: In-silico gene annotation prediction using the co-expression network structure. In: Cherifi, H., Gaito, S., Mendes, J.F., Moro, E., Rocha, L.M. (eds.) COMPLEX NETWORKS 2019. SCI, vol. 882, pp. 802–812. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-36683-4_64
25. Rust, A.G., Mongin, E., Birney, E.: Genome annotation techniques: new approaches and challenges. *Drug Discov. Today* **7**(11), S70–S76 (2002)
26. Valentini, G.: True path rule hierarchical ensembles. In: Benediktsson, J.A., Kittler, J., Roli, F. (eds.) MCS 2009. LNCS, vol. 5519, pp. 232–241. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02326-2_24
27. van Dam, S., Vösa, U., van der Graaf, A., Franke, L., de Magalhães, J.P.: Gene co-expression analysis for functional classification and gene-disease predictions. *Briefings Bioinform.* **19**(4), 139 (2017)
28. Vandepoele, K., Quimbaya, M., Casneuf, T., De Veylder, L., Van de Peer, Y.: Unraveling transcriptional control in *Arabidopsis* using CIS-regulatory elements and coexpression networks. *Plant Physiology* **150**(2), 535–546 (2009)
29. Yandell, M., Ence, D.: A beginner’s guide to eukaryotic genome annotation. *Nat. Rev. Genet.* **13**(5), 329–342 (2012)
30. Yon Rhee, S., Wood, V., Dolinski, K., Draghici, S.: Use and misuse of the gene ontology annotations. *Nat. Rev. Genet.* **9**(7), 509–515 (2008)
31. Zhou, G., Wang, J., Zhang, X., Guo, M., Yu, G.: Predicting functions of maize proteins using graph convolutional network. *BMC Bioinform.* **21**(S16), 420 (2020)
32. Zhou, Y., Young, J.A., Santrosyan, A., Chen, K., Yan, S.F., Winzeler, E.A.: In silico gene function prediction using ontology-based pattern identification. *Bioinformatics* **21**(7), 1237–1245 (2005)



Analysis of the San Luis Obispo Bay Microbiome from a Network Perspective

Viet Nguyen^(✉), Paul Anderson, Alexis Pasulka, and Theresa Migler

Cal Poly, San Luis Obispo, CA 93405, USA
{vnguy143,tmigler}@calpoly.edu

Abstract. Microorganisms are key players in ecosystem functioning. In this work, we describe steps to preprocess raw microbiome data, build a correlation network, and analyze co-occurrence patterns between microbes. We then apply the steps to a marine microbiome time-series dataset, collected over a year at the Cal Poly Pier. In analyzing this dataset, our goals include confirming known patterns of interactions and generating hypotheses about new patterns. Additionally, we analyse the co-occurrences between prokaryotic and eukaryotic taxa, which is rarely explored but can provide new insight into how marine microbial communities are structured and interact.

Keywords: Microbiome · Correlation network

1 Introduction

Microorganisms have a central place in marine biogeochemical processes, which control climate and ecosystem functioning [25]. In addition to interacting with the abiotic environment, microorganisms participate in complex biological interactions, which can have positive, negative, or neutral effects on their populations [7]. Detecting and understanding these relationships will help elucidate their impact on climate and ecosystem functioning. Microbiome datasets usually consist of abundances of operational taxonomic units (OTU) over many samples, and their pairwise relationships are based on the correlations between their abundance profiles, all of which form a correlation network. Correlation networks, however, are capable of more than just a way to visualize the interaction of OTUs. They can reveal modules of interconnected OTUs [5], detect relationship between communities and environmental conditions [5], quantify modularity of a community [3], and identify keystone OTUs (OTUs that have disproportionately large effects on their communities) [4].

This work is motivated by the lack of a clearly defined framework for preprocessing, constructing, and analyzing microbiome correlation networks. While there are common stages that researchers follow for microbiome analysis, tools for each stage are not standardized in their data format, and the knowledge about such tools are not centralized. Additionally, the San Luis Obispo (SLO) Bay, the sampling site of this dataset, is an understudied region, being analyzed

by only Barth et al. [1]; analysis from a correlation network perspective is novel for there and can give new insights to the marine microbiome of the bay.

Our contributions include the first network analysis of the SLO Bay’s microbiome, including a cross-domain analysis, and a detailed methods section for audience who do not have background in biology.

2 Background

Microbiome Data. The microbiome data being analyzed are generated through 16S and 18S rRNA gene high-throughput sequencing of the V4 region. Sequences have been clustered into operational taxonomic units (OTUs). Coupled with the microbiome data, the environmental variables including water temperature, chlorophyll *a*, oxygen, phosphate, nitrate, and silicic acid concentrations, and salinity. Each sample is a snapshot of the microbiome and its environment at a specific location at a specific time point. An OTU is a classification for a group of closely related organisms, often by similarity in gene sequences. For simplicity, OTUs can be thought of as “species”, although their precise taxonomic ranks can vary widely. Microbiome datasets usually come in raw sequence reads and are converted to relative abundance matrices, where each row represents an OTU and each column represents a sample. The value of each cell represents the abundance of the OTU relative to the sample. This form of data is called compositional data, because each column sums to 1.

Correlation Network. A node in a correlation network can be an OTU or an environmental factor (e.g. salinity, temperature). An edge between two OTUs or between an OTU and an environmental factor represents a covarying relationship between them. Edges can represent a positive or negative correlation and can have correlation coefficients as their weights.

Challenges in Constructing Robust Networks. Given a dataset, there are many ways to construct a correlation network through filtering OTUs and different correlation detection algorithms. The goal is to construct a network that is ecologically interpretable and represents the dynamics of the community. There are many challenges in doing so, including scale, compositionality bias, and data sparsity. Microbiome datasets can have over 16,000 OTUs, not only representing a challenge in correlation detection (which grows quadratically with the number of OTUs), but also the interpretability of the network. Because of the nature of compositional data, correlation measures can produce spurious correlations when the dataset has low diversity [8, 17].

3 Related Works

Many studies have applied correlation networks to study microbial communities, both in marine ecosystems and freshwater ecosystems. They have found modules of microbes that either have direct relationships (e.g. trophic) or share similar environmental preferences.

Chafee et al. studied the bacterioplankton community in a temperate coastal marine environment [5]. Using weekly sampled data over 2 years, they identified modules of co-varying microbes. The modules' abundances appear to be driven by oligotrophic and eutrophic conditions, which are controlled by seasonal phytoplankton blooms.

Cram et al. studied a marine bacterioplankton community at five layers of different depths: 5 m, the deep chlorophyll maximum layer (between 5 m and 40 m), 150 m, 500 m, and 890 m (just above the seafloor) [6]. Within the 2 uppermost layers and the deepest layer, they found two modules of microbes and environmental parameters where correlations are positive within each module and negative across modules indicating periodic shifts between two environmental conditions. Investigating time-lagged interactions between-depth, two-thirds of the correlations characterize a pattern of shallower population changes affecting deeper populations.

Jones et al. studied one year of monthly sampled data of bacteria and microbial eukaryotes at two locations within a low-salinity lake [13]. This is one of the few works in our literature review where both domains are studied and cross-domain correlation analysis is performed. The authors found correlation patterns that reflect the seasonality of the lake and disturbance events, including a bloom of the ecologically disruptive *P. parvum*.

While numerous works have indicated that environmental factors drive microbial communities' composition, short-term (daily to weekly) variability of marine protist communities studied by Berdjeeb et al. [3] cannot be fully explained by changes in their environment, but also by interspecific relationships, stochastic processes, mixing with adjacent water, and advection.

While correlation network analysis on microbiome data is not novel, the field is lacking a defined framework for approaching network analysis. Furthermore, the impact of difference methodological choices on the analysis and interpretation of microbiome networks has not been thoroughly investigated. Weiss et al. benchmarked different correlation detection strategies, a crucial choice in constructing a robust network, and make recommendations for their usage [28]. They also noted that while some techniques are better than others, each technique has their own caveat to be cautious of.

4 Materials and Methods

Sampling Site. Samples were collected as part of the Cal Poly Pier Microbial Oceanography Time-series, established by the Pasulka Lab in 2017 [1]. The Cal Poly Pier is located in the San Luis Obispo (SLO) Bay, which is an understudied part of the California Current System [1].

The coastal marine environment of central California is regulated by winds driving upwelling, a process where nutrient-rich water deep in the ocean is brought to replace the nutrient-depleted water at the coast [10]. This results in seasonal nutrient conditions that drive OTUs' abundances, as observed in previous works [5]. Peak Upwelling represents the period when this process takes

place most aggressively, when Upwelling Relaxation represents the period when winds that drive upwelling dwindle down or reverse.

In our dataset, there are three seasons: Winter Storms (spaning from December to February), Peak Upwelling (April-May), and Upwelling Relaxation (July-September). All of the other months are transitions between the seasons.

Dataset. Our dataset consists of sequence counts of 28 samples collected from the Cal Poly Pier from January 2018 to December 18 with the goal of capturing the seasonal dynamics of microbial communities. Following amplicon sequencing of both the prokaryotic and eukaryotic communities via 16S rRNA and 18S rRNA and OTU clustering, the dataset consists of 63,814 16S prokaryotic OTUs and 9,983 18S eukaryotic OTUs in 28 samples. The time gaps between samples range from one to five weeks.

The taxonomic hierarchy of prokaryotic OTUs includes seven levels: domain, phylum, class, order, family, and genus. The kingdom level is omitted from our data. Most of the 16S OTUs (75%) are identified to the most granular level. The taxonomic hierarchy of eukaryotic OTUs includes nine levels, and similar to the prokaryotic dataset the kingdom is omitted. Most of the 18S OTUs (83%) are identified to the most granular level.

In addition to the OTU counts, each sample has corresponding metadata: the temperature (in degree Celsius), salinity (gram of salt per kilogram of water), and Chlorophyll a concentration (microgram of Chlorophyll a per liter of water) of the water at the time and location the sample was taken.

Data Filtration. We filter OTUs as follows:

- For the seasonal network (16S Peak Upwelling), we retain OTUs that have relative abundance of at least 1% in at least one sample and are present in at least three samples, or half of the number of samples in each season as recommended by Weiss et al. [28].
- For year-long networks (16S, 18S, and cross-domain), we retain OTUs that have relative abundance of at least 1% in at least one sample and are present in at least three samples.

Correlation Detection. We choose to use SparCC [8] to detect correlation between OTUs based on their abundances, because it is appropriate for our dataset for many reasons. First, the dataset is unevenly sampled, i.e. the temporal gaps between samples are not equal, and Local Similarity Analysis (LSA) [20], a good metric for time series correlation, assumes the dataset is evenly sampled. Our dataset is sparse in the sense that there are many zero sequence counts (90% of the 16S and 85% of the 18S abundance tables' entries are zeros), and SparCC can deal with this by adding small pseudocounts (equivalent to Laplace smoothing). SparCC has also been shown to perform well on a diverse set of datasets [28]. Specifically, we use FastSpar (github.com/scwatts/FastSpar) [27], an faster implementation of SparCC with an unbiased P-value estimator.

The parameters that were used to run FastSpar on all subsets of our dataset is as follows: 50 iterations, 10 exclusion iterations, correlation strength exclusion

threshold of 0.1, and random number generator seed of 1. The p-values are calculated using bootstrapping with 1,000 random permutations, with 5 iterations per permutation.

For the year-long networks, we correct for multiple testing using the Benjamini-Hochberg procedure [2] using an alpha level of 0.05. For the seasonal networks, because of the limited number of samples, all of the correlations are not significant after the correction. Since this is a hypothesis generating study, we think that it is appropriate to accept a higher rate of false discovery, at an uncorrected alpha level of 001.

There are no weak correlations (with absolute value smaller than 0.8), which we would have filtered out.

Correlation with Environmental Variables and Cross-Domain. Aside from correlations between taxa, it can also be interesting to look at correlations between taxa and environmental variables. We use Spearman’s correlation instead of SparCC because environmental variables are not compositional with the sequence counts. Environmental variables are not transformed before correlation detection, but OTUs are normalized by the total sequence count of each sample.

Following Mikhailov et al. [18] and Jones et al. [13], we also use Spearman’s correlation to calculate correlations between prokaryotic OTUs and eukaryotic OTUs. It is not logical to concatenate the 16S sequence counts and 18S sequence counts and calculate SparCC correlations because prokaryotic and eukaryotic OTUs are sequenced separately. We thus normalize the two dataset separately (again, by the total sequence counts of each sample), and use the relative abundance vectors to calculate Spearman’s correlations (only between each prokaryotic and eukaryotic OTUs).

Graph Construction. We model the community as undirected, weighted graphs using the NetworkX library for Python [12]. Each graph represents the covariance relationships between taxa in a period of time. In each graph $G = (V, E)$, nodes represent taxa, and an edge between two taxa represents a correlation between them and has weight between -1 and 1 .

Initially, all nodes in the graph are OTUs. For effective visualization, we combine nodes based on their taxonomy and correlations, up to level 5 of the taxonomy. Level 5 roughly corresponds to the family level in most OTU’s taxonomies, although this may not be true for all OTUs. For each level from the OTU level to the 5th level, we group nodes by their taxonomy up to that level. For each such group, we check if (1) there are any negative correlations among the OTUs in the group and (2) if there are conflicting correlations with OTUs outside the group (e.g. OTU A within the group correlates positively with OTU C outside the group, but OTU B within the group correlates negatively with OTU C). We combine the group into a node if both conditions are false. If we do combine, a new “combined” node is created and will have edges between it and neighbors of the group where the weights of the new edges are the average of the weights of the edges between the external OTUs and the nodes in the group. For example, if the correlations between nodes A, B within the group and node C outside the group are 0.7, 0.9, then the correlation between the “combined”

node with C is $(0.7 + 0.9)/2 = 0.8$. Note that we may combine even when not all nodes in the group correlate with an external OTU, so this may overemphasize some patterns and should be kept in mind for analysis.

Clustering. To cluster the networks, we use *manta* (github.com/ramellose/manta) [21]. The graphs constructed with SparCC correlations are exported to the GML format. We run the *manta* script with the following configuration: minimum cluster size of 0, number of clusters from 2 to 10, and at least 90% of the edges must be used if the graph is not balanced. Because *manta* has a randomized component that may output different number of clusters and minimally different cluster assignments, we run the process 20 times for each graph and choose the majority result.

We run *manta* on the two year-long networks (16S and 18S) but not the seasonal networks (16S PU, 16S UR, 18S PU, 18S UR) because *manta* does not work with networks that contain low numbers of edges. We thus cluster the seasonal networks visually. The networks are imported into Cytoscape [23], and nodes are first arranged using the “Prefuse force directed layout”, where nodes that share positive edges are put close together, while nodes that share negative edges are pushed away from each other. Some initial clusters are formed by this layout, but we further examine each cluster and if a node is not too well connected with the rest of the cluster, we move it out to its own cluster.

Transitivity. We want to measure how significantly different the transitivity of these microbiome networks are from random networks created using Erdős-Rényi with the same number of nodes and edges. We do this by creating a distribution of 1000 such random networks’ transitivity, and calculate portion of the distribution that are greater than the corresponding real network’s transitivity.

5 Results

16S Peak Upwelling Network. The correlation network of prokaryotic taxa during Peak Upwelling season has primarily positive edges [1], which is consistent with the literature [3, 5, 16, 19, 26, 28]. The most connected nodes are the NS9 marine group, Polaribacter, and Tenacibaculum, each correlating with nine to ten other nodes. The transitivity is 0.52, meaning that there is a 52% probability that node B and C are connected if they are both connected with another node A. This indicates a well structured network and is significantly different from random networks of the same size ($p\text{-value} = 0.00$).

To analyze microbial networks, we first look at the five most abundant (colored yellow) and five most connected (colored red) taxa. The most abundant taxa are identified by the average relative abundance over the samples. There are no taxa that are both highly abundant and well connected, which is a notable pattern. This validates the accuracy of SparCC, because highly abundant taxa are likely to have spurious negative correlations [28].

OTUs of the NS9 marine group are the most connected, despite being low in abundance at only 0.13% over the season. It has negative correlations with

Table 1. Number of positive and negative edges for each network

Network	Positive edges	Negative edges	Edges	Nodes	Transitivity
16S Peak Upwelling	64 (85.33%)	11 (14.67%)	75	75	0.52
16S Upwelling Relaxation	2 (18.18%)	9 (81.82%)	11	59	0.00
16S Year-Long	146 (55.09%)	119 (44.91%)	265	54	0.51
18S Peak Upwelling	5 (50.00%)	5 (50.00%)	10	81	0.00
18S Upwelling Relaxation	0 (0.00%)	1 (100.00%)	1	14	0.00
18S Year-Long	121 (64.71%)	66 (35.29%)	187	50	0.44
Cross-Domain	132 (52.38%)	120 (47.62%)	252	77	0.00

OTUs of the Polaribacter and Ulvibacter genera. It was observed previously by Seo et al. [22] that the NS9 marine group tend to populate coastal environments (which is where the sampling site is) while Polaribacter and Ulvibacter reside in further out in the ocean. However this spatial variability does not explain the negative correlations observed over time at the same sampling site. Furthermore, abundances of Polaribacter and Ulvibacter at our coastal sampling site are equal to and greater than that of NS9, respectively (Table 1).

OTUs of the Polaribacter and Tenacibaculum covary positively and share 75% of their correlations with other taxa. Habib et al. previously found that some isolates of Tenacibaculum tend to phylogenetically cluster with a strain of Polaribacter [11]. Because the Tenacibaculum genus is known to include several fish pathogens, Habib et al. also speculated that pathogenic lineages are also in the Polaribacter genus, but a conclusion would require more analysis on complete genome data [11].

16S Year-Long Network. Correlations between prokaryotes over the year are shown in Fig. 1. There are 10% more positive edges than negative edges. Polaribacter and the NS9 marine group still have the most correlations similar to the Peak Upwelling network, but Synechococcus, which was not very connected during Peak Upwelling, has taken Tenacibaculum's place. Tenacibaculum does not appear in the year-long network because it does not meet the higher minimum abundance requirement. The transitivity is 0.51, similar to that of the Peak Upwelling network (p -value = 0.00).

manta is able to put the nodes into 3 clusters, where the majority are intra-cluster positive edges (green) and inter-cluster negative edges (orange), while there are few inter-cluster positive ones (blue) and intra-cluster negative ones (red). The relationships between the three clusters are structurally unbalanced, because the three clusters have negative relationships with each other. This goes against the assumption “the enemy of my enemy is my friend” that *manta* makes.

The most connected nodes are distributed among the three clusters. While it may be tempting to interpret them as keystone taxa, there are many caveats as discussed by Berry and Widder [4]. They identified four predictors of a keystone: high degree, low betweenness centrality, high closeness centrality, and high transitivity [4]. None of the nodes meet all of the above requirements, thus further experiments are needed before conclusion.

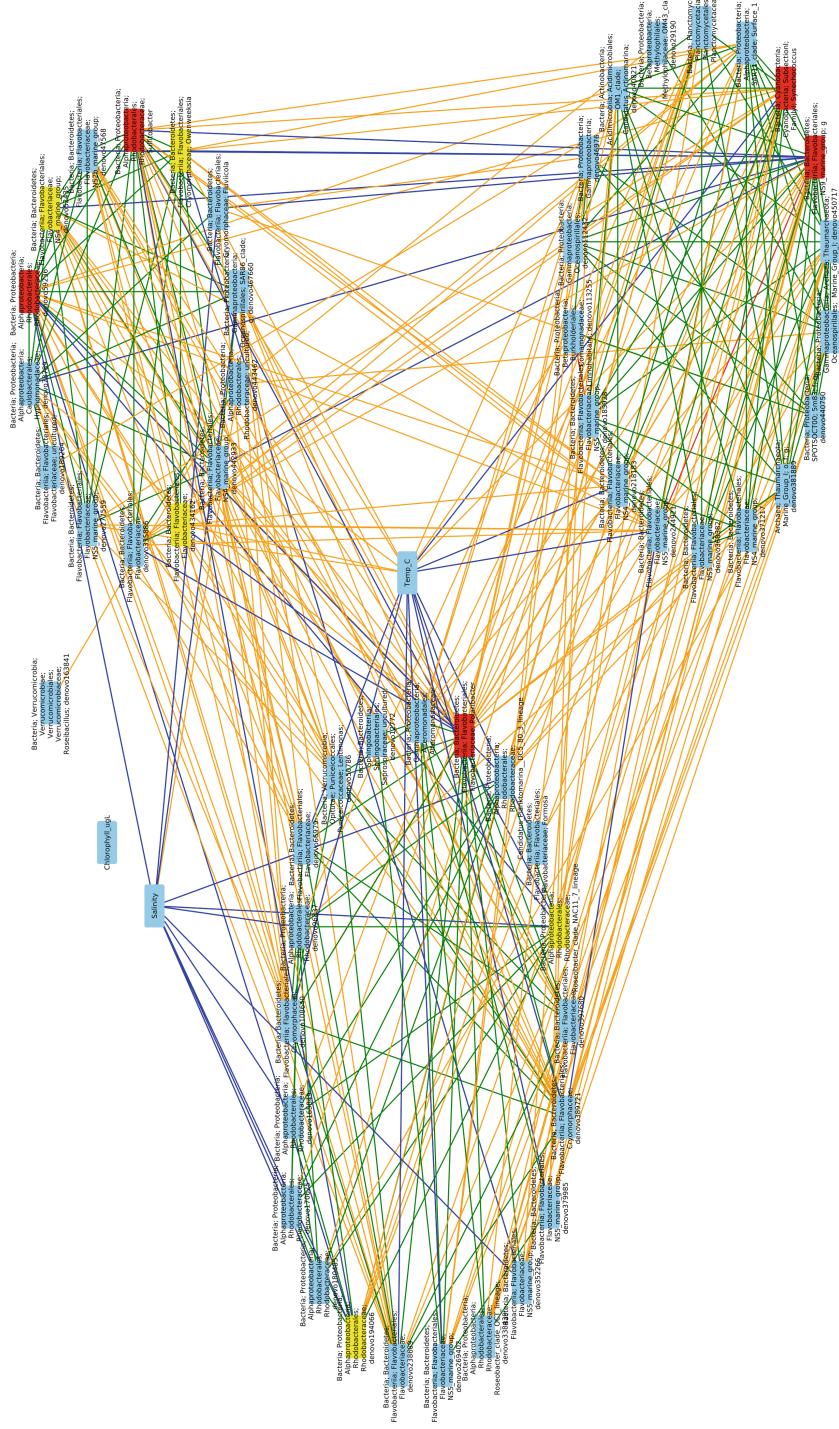


Fig. 1. Correlation network of 16S taxa over the entire year. Network is divided into 3 clusters as shown. Green edges are positive correlations within cluster, red edges are negative correlations within cluster, blue edges are positive correlations across clusters, and yellow edges are negative correlations across clusters. Red nodes are top connected taxa and yellow nodes are top abundant taxa.

18S Year-Long Network. While we built the seasonal 18S networks, each season was dominated by a few nodes and there is little dynamics. Thus those networks are unstructured, indicated by a transitivity of 0. Such a low transitivity is unusual for microbiome covariance networks [7]. We therefore only analyze the year-long network. The transitivity is 0.44 and is significantly different than from random networks of the same size (p-value = 0.00).

The clusters in this network are not as clear-cut as the ones in the 16S network, having multiple inter-cluster positive correlations and intra-cluster negative correlations. This is also exhibited by the high number of weak assignments by *manta*. Most of the nodes are dinoflagellates, and they also account for the most abundant nodes, as expected from the community composition analysis above. The only two diatom nodes are in the right cluster (Radial-centric-basal-Coscinodiscophyceae and Polar-centric-Medlophyceae). While the two nodes correlate positively, Medlophyceae is significantly more connected, and correlates negatively with dinoflagellates. Barth et al. observed a rotation in dominance between diatoms and dinoflagellates at the same sampling site [1], which explains this negative relationship.

The negative relationships within the right cluster are particularly interesting because they are associated with parasitic dinoflagellates, which are an important group that are often missed by microscopy studies.

One of the most connected taxa, Cryptomonadales, is positively correlated with three out of four taxa in the Ciliophora phylum (ciliates). A common genus among the Cryptomonadales lineages is *Teleaulax*, which is known to be a plastid donor to ciliates, [14] which may explain these positive relationships.

Cross-Domain Network. Microbial prokaryotes and eukaryotes also interact with each other, and previous studies have looked at this [13, 18]. The cross-domain network's transitivity is 0 not because there is a lack of structure, but because by definition, the network is bipartite (every edge must connect a prokaryote and a eukaryote) and thus there are no closed triplets.

Rhodobacteraceae has been noted to be a major vitamin supplier for eukaryotes, including dinoflagellates [24]. In our networks, several OTUs of the phylum Dinoflagellata correlate densely and positively with OTUs of the family Rhodobacteraceae, displaying this relationship. One OTU of the Ciliophora phylum (ciliates) also covaries with Rhodobacteraceae, and while we have not found previous literature on the relationship between these two taxa, Lee and Kugrans [15] have compared ciliates with dinoflagellates and found similarities between the two phyla's feeding apparatuses.

Discussion on Choosing Taxonomy Level for Analysis. There are no standard practices for choosing the taxonomy level to construct the network at. Different patterns emerge at different levels. Typically, it is desirable to differentiate organisms based on their ecological niche, but currently there is no consensus on the sequence similarity level (common values are 97%, 98%, and 99%) necessary to differentiate them [9]. Even more challenging is that the differentiating threshold varies for different lineages, such that two organisms whose

sequences are 97% similar can be ecologically identical while another pair of organisms that have 98% sequence similarity are ecologically distinct.

Existing studies at build their networks at the OTU level with sequence similarity ranging from 97% to 99% [3,5,6,13,18,19]. In our study, we have attempted to calculate correlations after collapsing OTUs to the genus and family levels. Despite finding significant correlations between the taxa at those levels, we found little ecologically interesting patterns in the network. Subsequently we also found that there is a lack of discussion on whether building networks at higher taxonomic levels was valid or would reveal patterns unobservable otherwise. Thus we follow the literature and built our networks at the OTU level.

Calculating correlations at the OTU level can be challenging for practical and statistical reasons. Pairwise correlations detection necessitates quadratic algorithms, making network construction for such large datasets impractical. Downstream tasks such as module detection and analysis are also impractical. From a statistics perspective, each pairwise correlation is a hypothesis test and thus requires correction for multiple testing. With so many tests, the corrected α with Bonferroni correction would be effectively 0 and p-values corrected by the less conservative Benjamini-Hochberg procedure [2] are too large to be rejected.

In this study, we still detect correlations at the OTU level, but to address the computation and statistics concern, we filtered out OTUs that are not abundant and prevalent enough to have large effects on the communities. To make networks easier to be analyzed visually, we combine nodes with the same taxonomy and similar interactions afterwards. However, combining nodes iteratively without a stopping point can lead to nodes that only have one or two taxonomy levels (in our case, Archaea was one such node in the 16S network) and thus are meaningless for analysis. We therefore only combine nodes up to the Family level.

6 Conclusion

In this paper, we describe the steps and consideration that we and most studies take to construct a correlation network. Our intent is to fill the gap that will help novices in the field. We construct and analyze a novel microbiome dataset of the SLO Bay and found ecologically interesting patterns. We also analyzed the correlations between organisms of different domains, an understudied portion of the field.

References

1. Barth, A., Walter, R., Robbins, I., Pasulka, A.: Seasonal and interannual variability of phytoplankton abundance and community composition on the central coast of California. *Marine Eco. Progress Series* **637**, 29–43 (2020)
2. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Statist. Soc. Ser. B* **57**(1), 289–300 (1995). [http://links.jstor.org/sici?&sici=0035-9246\(1995\)57:1289:CTFDRA2.0.CO;2-E&origin=MSN](http://links.jstor.org/sici?&sici=0035-9246(1995)57:1289:CTFDRA2.0.CO;2-E&origin=MSN)

3. Berdjeb, L., Alma, E., Needham, D., Fuhrman, J.: Short-term dynamics and interactions of marine protist communities during the spring-summer transition. *ISME J.* **12**(8), 1907–1917 (2018)
4. Berry, D., Widder, S.: Deciphering microbial interactions and detecting keystone species with co-occurrence networks. *Front. Microbiol.* **5**, 219 (2014)
5. Chafee, M., et al.: Recurrent patterns of microdiversity in a temperate coastal marine environment. *ISME J.* **12**(1), 237–252 (2017)
6. Cram, J., Xia, L., Needham, D., Sachdeva, R., Sun, F., Fuhrman, J.: Cross-depth analysis of marine bacterial networks suggests downward propagation of temporal changes. *ISME J.* **9**(12), 2573–2586 (2015)
7. Faust, K., Raes, J.: Microbial interactions: from networks to models. *Nat. Rev. Microbiol.* **10**, 538–550 (2012)
8. Friedman, J., Alm, E.: Inferring correlation networks from genomic survey data. *PLoS computational biology* **8**, e1002687 (2012)
9. Fuhrman, J., Cram, J.A., Needham, D.: Marine microbial community dynamics and their ecological interpretation. *Nat. Rev. Microbiol.* **13**, 133–146 (2015)
10. García-Reyes, M., Largier, J.L.: Seasonality of coastal upwelling off central and northern California: new insights, including temporal and spatial variability. *J. Geophys. Res.* **117**(C3) (2012). <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2011JC007629>
11. Habib, C., et al.: Multilocus sequence analysis of the marine bacterial genus *Tenacibaculum* suggests parallel evolution of fish pathogenicity and endemic colonization of aquaculture systems. *Appl. Environ. Microbiol.* **80**(17), 5503–5514 (2014)
12. Hagberg, A.A., Schult, D.A., Swart, P.J.: Exploring network structure, dynamics, and function using networkx. In: Varoquaux, G., Vaught, T., Millman, J. (eds.) *Proceedings of the 7th Python in Science Conference*, pp. 11–15. Pasadena, CA USA (2008)
13. Jones, A., Hambright, K., Caron, D.: Ecological patterns among bacteria and microbial eukaryotes derived from network analyses in a low-salinity lake. *Micro. Ecol.* **75**(4), 917–929 (2018)
14. Kim, J.I., Yoon, H., Yi, G., Kim, H.S., Yih, W., Shin, W.: The plastid genome of the cryptomonad *teleaulax amphioxeia*. *PLoS ONE* **10**(6), e0129284 (2015)
15. Lee, R., Kugrens, P.: Relationship between the flagellates and the ciliates. *Microbiol. Rev.* **56**(4), 529–42 (1992)
16. Liu, J., Meng, Z., Liu, X., Zhang, X.H.: Microbial assembly, interaction, functioning, activity and diversification: a review derived from community compositional data. *Marine Life Sci. Technol.* **1**(1), 112–128 (2019)
17. Lovell, D., Muller, W., Taylor, J., Zwart, A., Hellwell, C.: Caution! compositions! can constraints on omics data lead analyses astray? (2010)
18. Mikhailov, I., et al.: Co-occurrence networks among bacteria and microbial eukaryotes of lake Baikal during a spring phytoplankton bloom. *Microbial Ecol.* **77**(1), 96–109 (2019)
19. Needham, D., Sachdeva, R., Fuhrman, J.: Ecological dynamics and co-occurrence among marine phytoplankton, bacteria and myoviruses shows microdiversity matters. *ISME J.* **11**(7), 1614–1629 (2017)
20. Ruan, Q., Dutta, D., Schwalbach, M.S., Steele, J.A., Fuhrman, J.A., Sun, F.: Local similarity analysis reveals unique associations among marine bacterioplankton species and environmental factors. *Bioinformatics* **22**(20), 2532–2538 (2006). <https://doi.org/10.1093/bioinformatics/btl417>

21. Röttjers, L., Faust, K.: Manta - a clustering algorithm for weighted ecological networks. *Msystems* **5**(1), e00903-19 (2019)
22. Seo, J.H., Kang, I., Yang, S.J., Cho, J.C.: Characterization of spatial distribution of the bacterial community in the south sea of Korea. *PLOS ONE* **12**(3), 1–18 (03 2017). <https://doi.org/10.1371/journal.pone.0174159>
23. Shannon, P., Markiel, A., Ozier, O., Baliga, N.S., Wang, J.T., Ramage, D., Amin, N., Schwikowski, B., Ideker, T.: Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* **13**(11), 2498–2504 (2003)
24. Simon, M., Scheuner, C., Meier-Kolthoff, J.P., Brinkhoff, T., Wagner-Döbler, I., Ulbrich, M., Klenk, H., Schomburg, D., Petersen, J., Göker, M.: Phylogenomics of rhodobacteraceae reveals evolutionary adaptation to marine and non-marine habitats. *ISME J.* **11**, 1483–1499 (2017)
25. Singh, B., Bardgett, R., Smith, P., Reay, D.: Microorganisms and climate change: terrestrial feedbacks and mitigation options. *Nat. Rev. Microbiol.* **8**, 779–90 (2010)
26. Trombetta, T., Vidussi, F., Roques, C., Scotti, M., Mostajir, B.: Marine microbial food web networks during phytoplankton bloom and non-bloom periods: Warming favors smaller organism interactions and intensifies trophic cascade. *Front. Microbiol.* **11** 2657 (2020). <https://www.frontiersin.org/article/10.3389/fmicb.2020.502336>
27. Watts, S.C., Ritchie, S.C., Inouye, M., Holt, K.E.: FastSpar: rapid and scalable correlation estimation for compositional data. *Bioinformatics* **35**(6), 1064–1066 (2018). <https://doi.org/10.1093/bioinformatics/bty734>
28. Weiss, S., et al.: Correlation detection strategies in microbial data sets vary widely in sensitivity and precision. *ISME J.* **10**(7), 1669–1681 (2016)



From Quantitative SBML Models to Boolean Networks

Athénaïs Vaginay^{1,2(✉)}, Taha Boukhobza¹, and Malika Smaïl-Tabbone²

¹ Université de Lorraine, CNRS, CRAN, 54000 Nancy, France
athenais.vaginay@loria.fr

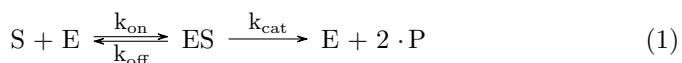
² Université de Lorraine, CNRS, Inria, LORIA, 54000 Nancy, France

Abstract. Modelling complex biological systems is necessary for their study and understanding. SBML is the standard format to represent models of biological systems. Most of the curated models available in the repository Biomodels are quantitative, but in some cases qualitative models—such as Boolean networks—would be better suited. This paper is the first to focus on the automatic transformation of quantitative SBML models to Boolean networks. We propose SBML2BN, a pipeline dedicated to this task. By running SBML2BN on more than 200 quantitative SBML models, we provide evidence that we can automatically construct Boolean networks which are compatible with the structure and the dynamics of a given quantitative SBML model.

Keywords: Boolean networks · Model transformation · SBML · Systems biology

1 Introduction

Life is based on biological systems which are essentially composed of biological components (genes, proteins, metabolites) acted upon by processes. However, they are highly complex, as molecular abundances and interactions change over time in response to external stimuli as well as to dynamical intra-system processes. The biological system that serves as a running example throughout this article is an enzymatic process: first, an enzyme E reversibly binds a molecule of substrate S (reactions \mathcal{R}_{on} and \mathcal{R}_{off}); together, they form the complex ES; then the substrates are transformed into two molecules of a product P while E returns to its free state (reaction \mathcal{R}_{cat}). The classical chemical notation of this system is:



Each of the three reactions is represented by an arrow from the *reactants* (i.e., components consumed during the reaction) to the *products* (i.e., components created during the reaction). On top (or below) of the arrow is the *speed constant*, which is a proportionality coefficient between the *stoichiometry* (amount) of the reactants and the rate of the reaction.

Through simulations, dynamic models of biological systems are of particular interest because they are useful proxies to understand and predict the behaviour and the dynamics of biological systems. For example, one can study how the speed of production of the product P is affected by the presence of an inhibitor of the enzyme E. However, the quality of the predictions strongly depends on the quality of the model, which in turn strongly depends on the quality of the data and the depth of the knowledge used to build the model. Moreover, biological models are often hand-crafted and this is error-prone. The repository Biomodels contains a peer-reviewed and curated collection of over a thousand models [20]. Models in Biomodels are described in the Systems Biology Markup Language (SBML), which is the most widely used standard representation language in the field of system biology. Several modelling formalisms exist, ranging from detailed ones (such as differential equations) to the most simple ones (such as Boolean networks). Most SBML models in Biomodels are quantitative models (mostly differential equations). However, in some cases, qualitative models like Boolean networks are more suited [3]. Indeed, their simplicity make them easy to study. They are in particular easily amenable to model checking [16] and control [3], even for large models.

In this paper, we propose SBML2BN, an *automatic* pipeline to synthesise a set of Boolean networks (BNs) modelling a biological system, starting from an SBML representation of the system. First, we introduce the key notions about Boolean networks and the principles of their synthesis starting from the given structure and dynamics of the biological system under study (Sect. 2). Then, we present the pipeline SBML2BN and detail its four steps (Sect. 3): (i) we extract the structure and (ii) the dynamics of the biological system from the SBML model; (iii) we use this information as constraints for the synthesis of the BNs; (iv) we assess the quality of the BNs produced by quantifying how well they fit to the structure and the dynamics of the input SBML model. We also give details about the pipeline implementation, which reuses and extends several published methods and software packages. Finally, we report the evaluation of SBML2BN by running it on more than 200 curated SBML models from the Biomodels database (Sect. 4). We provide evidence that the resulting BNs are in line with the biological system under study. We close the paper with conclusions and a few perspectives.

2 Boolean Networks and Their Synthesis

2.1 Definitions

Boolean networks (BNs) were introduced by Kauffman [14] and Thomas [25] to model genetic regulatory networks. Concepts used in BNs are described in a recent review [24]. An example of BN is given in Fig. 1 and used to illustrate the concepts introduced in the following.

The *components* of a BN are the components of the considered biological system. For example, the BN \mathcal{B}_1 (Fig. 1) has four components: S, P, E and ES. A *configuration* of a BN is a vector that associates a Boolean value ($\mathbb{B} = \{0/\text{inactive}; 1/\text{active}\}$) to each of the n components of the BN (in alphabetical order). For example, in the configuration 0000, no components is active, while only E is active in the

configuration 1000. A BN with n components has 2^n possible configurations. Each component X has an associated *transition function* $f_X : \mathbb{B}^n \rightarrow \mathbb{B}$ that maps the configurations of the BN to the next value of the component. The transition functions are usually written as Boolean expressions. In this paper, these expressions are in Disjunctive Normal Form (DNF), i.e., disjunctions of conjunctions. Moreover, the conjunctions are *satisfiable*, i.e., they do not contain both a literal and its contrary. The operators \neg, \wedge, \vee represent respectively negation, conjunction and disjunction. The transition function $f_{ES} := (E \wedge \neg S) \vee (\neg E \wedge S)$ states that the value of ES will be 1 if either the value of E or of S was 1 in the previous configuration. Figure 1a shows examples of transition functions with only one term.

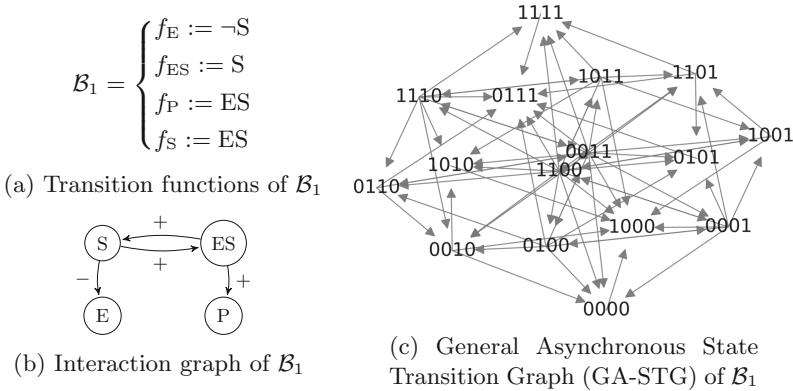


Fig. 1. Example of a possible Boolean network (among others) to model Eq. (1)

The structure of a BN is defined in terms of parent-child relationships between the components. A component P that appears in the transition function of a component X is called a *parent* of X . If the parent P is negated in the DNF associated with X , we say that the *polarity* of the influence of P on X is negative. Conversely, if the parent is not negated, this influence is positive. The *Interaction Graph* (IG) summarises these relationships as a directed graph. The directed edge $P \rightarrow X$ is labelled with “+” or “-” depending on the polarity of the influence P has on X . The interaction graph of \mathcal{B}_1 (Fig. 1b) contains the edges $S \xrightarrow{-} E$ and $S \xrightarrow{+} ES$ because S appears negatively in the transition function of E and positively in the one of ES . We will see in Sect. 2.2 how the IG is used to define the compatibility of a BN towards a given structure.

The BN *dynamics* is obtained by applying iteratively the transition functions starting from all possible configurations. The order of application of the transition functions is defined by the *update scheme*. The most common are the *synchronous*, *asynchronous* and *general asynchronous*. In the synchronous update scheme, the transition functions are applied all at once, while in the asynchronous update scheme, they are applied one by one (non-deterministically). In the general

asynchronous update scheme, any number of components can be updated at each step. Thus, it includes the updates possibilities of both the synchronous and asynchronous update schemes. The *state transition graph* (STG) is a directed graph whose nodes are the 2^n possible configurations of the BN. It contains a directed edge from c to c' if c' is the result of applying on c the transition function(s) according to the chosen update scheme. Figure 1c shows the General-Asynchronous STG (GA-STG) of \mathcal{B}_1 (Fig. 1a). We will see in Sect. 2.2 how the presence of specific edges in the GA-STG of a BN is used to measure the compatibility of this BN towards a given dynamics.

2.2 Synthesis of BNs Compatible with a Structure and a Dynamics

In general, a Boolean network that models a biological system has to satisfy two categories of *constraints*. On one hand, its structure has to comply with what is known on the system's structure. This knowledge concerns the list of components (genes, proteins...) involved and how they influence each others. Influences have a *polarity*: activation (polarity “+”) or inhibition (polarity “−”). The *parents* of a component X are the components which are known to influence X. A *Prior Knowledge Network* (PKN) encodes such knowledge. The nodes of the network are the components of the system, and directed edges parent → child are labelled “+” or “−” according to the polarity of the influences. Figure 2b shows an example of PKN of the enzymatic reaction Eq. (1). In this PKN, S, ES and E are the parents of E with polarities “−”, “+” and “−”. The PKN is used to constrain the structure of the synthesised BNs: a BN is *compatible* with a given PKN if its interaction graph is a spanning subgraph of the PKN. In other words, the interaction graph of a BN compatible with a given PKN is formed from the nodes and a subset of the edges of the PKN. This results in constraining which components can appear as variables in each transition function and the polarity of those variables. Hence, a component P is allowed in the transition function of a component X with a polarity σ if the PKN contains an edge $P \xrightarrow{\sigma} X$. For example, \mathcal{B}_1 (Fig. 1a) is compatible with the PKN given in Fig. 2b. On the contrary, a Boolean network having the transition function $f_E := \neg S \vee \neg ES$ is not compatible. Indeed, despite ES being a possible parent of E, the negative polarity is not allowed since $ES \xrightarrow{-} E$ is not in the PKN.

On the other hand, the dynamics of the BN has to comply with what is known on the system's dynamics. Starting from a given multivariate Time Series (TS) of the concentrations of the components over time, we can extract a sequence of configurations by binarising the TS. For example, the sequence of configurations extracted from the binarisation of the multivariate TS given in Fig. 2c is $0011 \rightarrow 1011 \rightarrow 1010 \rightarrow 1000 \rightarrow 1100 \rightarrow 0101$. Ideally, we would like this sequence to be a *walk* in the General Asynchronous STG (GA-STG) i.e., that the GA-STG contains all the edges appearing in the sequence. In such a case, the *coverage ratio* of the GA-STG towards the configuration sequence (defined as the number of edges present in the graph divided by the number of distinct edges in the sequence) is of 1 and the Boolean network is said to be *fully compatible* with the multivariate TS.

However, it is not always possible to retrieve the complete walk in the GA-STG [23]. In this case, the goal is to have the best coverage ratio possible.

All in one, a Boolean network is *compatible* with a Prior Knowledge Network (PKN) if its interaction graph is a subgraph of the PKN, and the compatibility between a Boolean network and a multivariate Time-Series (TS) is quantified using the *coverage ratio*. An ideal Boolean network synthesis method constructs *only* Boolean networks compatible with the given PKN and with *maximal* coverage ratio (of 1) in regard of the given multivariate TS.

$$\begin{aligned} \frac{d[E]}{dt} &= -k_{on}[E][S] + k_{off}[ES] + k_{cat}[ES] \\ \frac{d[ES]}{dt} &= k_{on}[E][S] - k_{off}[ES] - k_{cat}[ES] \\ \frac{d[P]}{dt} &= 2k_{cat}[ES] \\ \frac{d[S]}{dt} &= -k_{on}[E][S] + k_{off}[ES] \end{aligned} \quad \begin{aligned} k_{on} &= 1 \times 10^6 \text{ L mol}^{-1} \text{ s}^{-1} \\ k_{off} &= 0.2 \text{ s}^{-1} \\ k_{cat} &= 0.1 \text{ s}^{-1} \\ [ES]_0 &= 0 \text{ mol L}^{-1} \\ [P]_0 &= 0 \text{ mol L}^{-1} \\ [S]_0 &= 1 \times 10^{-6} \text{ mol L}^{-1} \\ [E]_0 &= 5 \times 10^{-7} \text{ mol L}^{-1} \end{aligned}$$

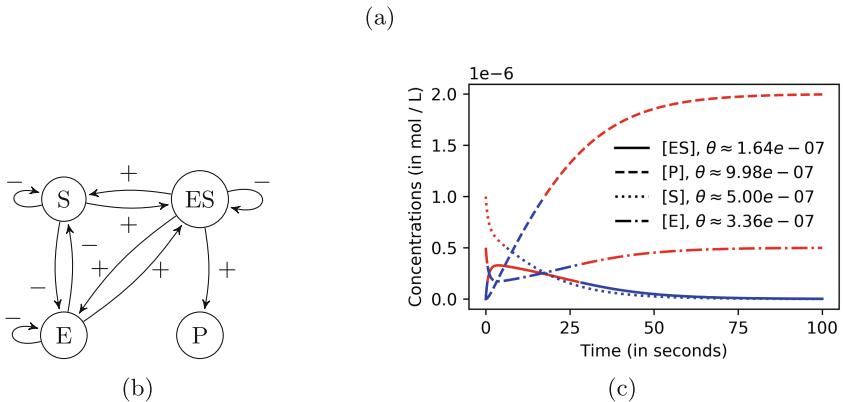


Fig. 2. (a) ODE system and its parametrisation, and (b) prior knowledge network for Eq. (1). (c) shows the multivariate time series, binarisation thresholds and resulting binarisation (blue if 0 and red if 1) obtained by simulation of the ODE system.

3 Description of the SBML2BN Pipeline

We propose SBML2BN, a pipeline for the automatic synthesis of Boolean networks starting from an existing quantitative SBML description of a biological system. All the necessary concepts about SBML are described in Sect. 3.1. The structure (PKN) and the dynamics (TS) of the biological system under study are extracted from the given SBML (Sects. 3.2 and 3.3). In the BN synthesis step (Sect. 3.4), the former is used to hard constrain the structure of the resulting BNs, while the latter acts as soft constraints. The pipeline finishes with the evaluation of the *set* of achieved BNs (Sect. 3.5).

3.1 Complete Quantitative SBML Models in a Nutshell

The Systems Biology Markup Language (SBML) [15] is an XML markup language. The SBML file representing the biological system from Eq. (1) is given in the GitLab repository associated to this paper[†]. The SBML standard¹ specifies how an SBML file is structured and how the different elements are named. This paper focuses on a subset of SBML models which contains all the necessary information for the SBML2BN pipeline to process the model. We refer to these SBML models as *complete quantitative SBML models*. We describe the content of such models as follows.

The biological components involved in the biological system (referred to as *species* in SBML) are supplied, as well as their initial concentration. The reactions taking place in the biological system are described. The definition of a reaction \mathcal{R} is composed of a list of *reactants*, a list of *products*, and a *kinetic law* $e_{\mathcal{R}}$ (i.e., a mathematical expression which gives the rate of the reaction \mathcal{R}). For each species X involved in a reaction \mathcal{R} , the *net stoichiometry* $\nu_{\mathcal{R}}^X$ of X in \mathcal{R} is the amount of X as a product minus its amount as a reactant. If $\nu_{\mathcal{R}}^X > 0$ (resp. < 0), X is *effectively produced* (resp. consumed) by the reaction \mathcal{R} . Sometimes, a reaction involves *modifiers*, i.e., species which influence the speed of the reaction without having their amount modified. A modifier which increases (resp. decreases) the speed of the reaction is an *activator* (resp. *inhibitor*). Finally, all the supplementary kinetic parameters and their values are specified.

3.2 Extraction of the PKN from the SBML Model

This first step consists in the construction of the PKN (noted \mathcal{G}). Figure 2b is the PKN constructed by SBML2BN for Eq. (1). The PKN we obtain in this step corresponds to the Syntactical Influence Graph (SIG) of an SBML model [11]. The nodes of the PKN are the SBML species of the SBML model. As for the edges, they are obtained by applying the following rules on each reaction of the SBML model:

- **If** X is a reactant or an activator and Y disappears **then** $X \xrightarrow{-} Y \in \mathcal{G}$
- **If** X is an inhibitor and Y appears **then** $X \xrightarrow{-} Y \in \mathcal{G}$
- **If** X is a reactant or an activator and Y appears **then** $X \xrightarrow{+} Y \in \mathcal{G}$
- **If** X is an inhibitor and Y disappears **then** $X \xrightarrow{+} Y \in \mathcal{G}$

As detailed in Sect. 3.1, a modifier can be either an activator or an inhibitor (or both). In some SBML models, specific annotations (using the System Biology Ontology [7]) indicate the exact role of the modifiers. When such annotations are missing, the modifier is considered as both activator and inhibitor.

¹ <http://sbml.org/Documents/Specifications>.

3.3 Extraction of the Time-Series from the SBML Model

The goal of this step is to retrieve the concentrations of the species over time. Since the processed SBML model is *complete*, it contains all the necessary information to construct a working Ordinary Differential Equations system (ODE). To do so, an expression representing the overall rate of change of the amount of each species is constructed as the sum of the contributions of all the *relevant* reactions (i.e., reaction in which a given species is involved as a product or a reactant). For example, in the running example, the species ES is involved as a product in reaction \mathcal{R}_{on} and as a reactant in reactions \mathcal{R}_{cat} and \mathcal{R}_{off} . Hence, the overall rate of change of ES is: $\frac{d\text{ES}}{dt} = \nu_{\mathcal{R}_{\text{on}}}^{\text{ES}} \cdot e_{\mathcal{R}_{\text{on}}} + \nu_{\mathcal{R}_{\text{off}}}^{\text{ES}} \cdot e_{\mathcal{R}_{\text{off}}} + \nu_{\mathcal{R}_{\text{cat}}}^{\text{ES}} \cdot e_{\mathcal{R}_{\text{cat}}}$ with $\nu_{\mathcal{R}_{\text{on}}}^{\text{ES}} = 1$, and both $\nu_{\mathcal{R}_{\text{off}}}^{\text{ES}}$ and $\nu_{\mathcal{R}_{\text{cat}}}^{\text{ES}} = -1$. The SBML representation[†] of Eq. (1) indicates that the speed of each reaction is proportional (with a factor k_R) to the product of the amount of reactants. Figure 2a shows the ODE system, parameterisation and initial conditions retrieved from the SBML file[†].

We then run a deterministic numerical time integration from $t = 0$ to t_{\max} of the ODE system. Figure 2c shows the multivariate TS by simulating Fig. 2a for $t_{\max} = 100$ s (chosen arbitrarily).

3.4 Boolean Networks Synthesis

At this stage, the goal is to construct automatically a set of BNs compatible with the PKN and the multivariate TS. The synthesis problem is largely under-specified, since only one multivariate TS is provided. Several methods have been dedicated to this task [18, 19, 22]. They exploit various strategies, in particular regarding the fitting of the transition functions to given multivariate TS.

In [26], we introduced ASKeD-BN and showed that it is the best synthesis method available in the case of signed PKN and complete multivariate TS (i.e., without missing time steps). ASKeD-BN exhaustively synthesises BNs compatible with a given PKN and multivariate TS with respect to two criteria that correspond closely to the notion of *compatibility* defined in Sect. 2.2:

1. The interaction graph of the synthesised BNs are compatible with the given PKN (i.e., be a subgraph of the PKN), and they have the smallest number of edges possible.
2. The dynamics of each component minimises the mean absolute error with regard to the multivariate TS.

The choice of the binarisation might be crucial for the outcome. ASKeD-BN uses the simplest procedure possible: a threshold θ_X is chosen for each component X as $\min + (\max - \min)/2$, where min and max are the observed minimum and maximum of X in the time series. With x_t the value of the concentration of the species X at time t, the binarised value of X at time t is 1 if $x_t \geq \theta_X$ and 0 otherwise. In the multivariate TS (Fig. 2c), the red (resp. blue) parts of the lines correspond to concentration values which are bigger (resp. small) than the corresponding threshold, hence will result in 1 (resp. 0). After binarisation, we can extract the following configuration sequence: 0011 → 1011 → 1010 → 1000 → 1100 → 0101.

ASKeD-BN fits the transitions functions by using the mean absolute error to penalise the candidates for each transition they cannot explain.

Other methods such as caspo-TS [22] work on explaining the reachability of the configurations. Hence, wildcard are added to the configurations sequence: $0011 \rightarrow * \rightarrow 1011 \rightarrow * \rightarrow 1010 \rightarrow \dots$. This feature is an asset in the case of missing time points, but here, the multivariate TS are complete, and this feature is not necessary (and even counter-productive [26]).

3.5 Evaluation of Synthesised Boolean Networks

In this last step, we evaluate the compatibility (such as defined in Sect. 2.2) of all the Boolean networks synthesised by the SBML2BN for the input SBML model. Since they are compatible with the PKN (by construction), our quality check focuses on the compatibility with the multivariate TS: we compute the *coverage ratio* of each BN, and we aggregate the individual coverage ratios using the median and standard deviation. Ideally, the pipeline would return *only* BNs with *maximal* coverage ratios.

3.6 Pipeline Implementation

We have made a point of supporting reproducibility and facilitating the installation of the different tools. All the tools developed and reused are open-source, well documented and freely available. The pipeline is managed using Snakemake [21] (which ensures each step is ran properly and in the correct order) and installed using Conda [1] (which simplifies the management of library dependencies and avoid version conflicts). We use the parser libSBML [4] to retrieve the PKN, COPASI [13] to retrieve the multivariate TS, and PyBoolNet [17] to compute the AG-STG of the BNs. For the BN synthesis step, we used a declarative implementation of ASKeD-BN [26], which uses Answer-Set Programming [12].

4 Evaluation of the SBML2BN Pipeline

4.1 Evaluation on the Running Example Eq. (1)

We apply SBML2BN on the SBML file[†] modelling Eq. (1). The Boolean network \mathcal{B}_1 (Fig. 1a) is the only solution we obtain. Its interaction graph (Fig. 1b) is a spanning subgraph of the PKN. and its GA-STG (Fig. 1c) covers all the 5 transitions extracted from the binarised TS. Its coverage ratio is thus 1, and the coverage median and standard deviation of this singleton of solutions are obviously 1 and 0 respectively, making SBML2BN plainly successful.

4.2 Evaluation on SBML Models from BioModels

BioModels [20] is a repository of models of biological and biomedical systems, including metabolic networks, signalling networks, gene regulatory networks and

infectious diseases. All models stored in the curated branch of BioModels are encoded in SBML and have passed a drastic manual curation process, which asserts that the simulations from the paper in which the model was originally published are reproducible by the SBML model. The latest available release of Biomodels² contains 640 SBML curated models, including 369 complete quantitative SBML (i.e., models for which SBML2BN is able to extract a PKN and a multivariate TS). However, the complexity of the BN synthesis problem increases exponentially with the number of parents for each component. Indeed, the number of possible transition functions for a component with p parents is 2^{2^p} . We assume the problem is not tractable if a component has more than 10 parents, hence we only make the evaluation onto the 209 SBML models that have all their components with less than 10 parents. The number of components in these models ranges from 2 to 60, but bigger models would not have been a problem since ASKeD-BN is not directly impacted by the number of components. For each SBML model, the length of simulation (t_{\max}) is extracted by hand from the curation reports of Biomodels.

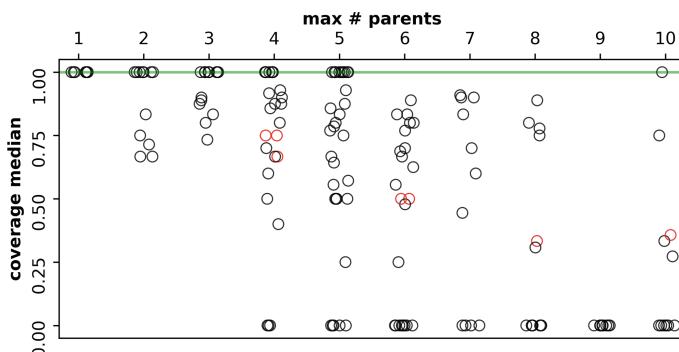


Fig. 3. Coverage evaluation for the BNs synthesised by SBML2BN for 155 SBML models. Each dot represents the set of BNs returned for a given SBML model. Its coordinates are the coverage ratio median (ordinate) and the number maximal of parents for the nodes of the SBML model (abscissa). Sets of BNs having a coverage variance strictly positive are represented by red dots. Green line shows where are the dots when the pipeline only returns BNs with a perfect coverage.

The pipeline processes about three fourth (155) of the models in less than 30 h (median of 36 min—data not shown). In the following, we report the results for these models. In average, the pipeline synthesises 6.5 Boolean networks per SBML. This number masks a strong disparity, since a single BN was synthesised for more than half (106) SBML models. Figure 3 summarises the coverage evaluation. As said before, all the BNs returned by the pipeline for a given SBML model would ideally have a perfect coverage ratio, hence with a median of 1 and a standard deviation of 0. The pipeline synthesises only perfect BNs for one fourth (39) SBML models. The median and standard deviation of the median coverage ratios of the BNs

² Release 31 <ftp://ftp.ebi.ac.uk/pub/databases/biomodels/releases/2017-06-26/>.

synthesised for a given SBML model are 0.77 and 0 respectively. They are only 12 models (in red in Fig. 3) for which the standard deviation is not 0 (range 0.07; 0.25). Overall, the pipeline is efficient at finding Boolean networks with good coverage median and small standard deviation. Nevertheless, we can significantly correlate a loss of performance to the maximum number of parents in the systems (Kendall τ value of -0.43 , p-value of $1.51e-13$). We are currently investigating reasons of this correlation. One reason could simply be that Boolean networks cannot explain all phenomena (Sect. 2.2): in some cases, the maximum achievable coverage ratio is smaller than 1, but our quality evaluation of the synthesised BNs does not take this fact into account. We could use Boolean networks with the most-permissive semantics [5] to overcome this limitation, but no implementation is available for BNs having non-monotonous transition functions (such as the ones our pipeline is likely to produce). Another reason could be that the specification of SBML leaves open the possibility for a model to contain contradictory information. It has been showed in [9] that more than 60% of the SBML models tested in 2012 were containing contradictions. Among the contradictory models they identified, the model n°44³ has reactions with components used in the kinetics which are not listed as reactants nor modifiers. This has a bad impact on the construction of the PKN by our pipeline (Sect. 3.2), since potential parents of some components are not identified as such. For this model, one BN was generated, with a not so good coverage of 0.55. We are planning to investigate how to remove beforehand the contradictions from these models [9].

5 Conclusion and Perspectives

In this paper, we presented SBML2BN, a pipeline for the automatic transformation of a *complete* quantitative SBML model into a set of compatible Boolean networks. The transformation of biological models from a formalism to another has been investigated in several papers [2, 10] in particular from ODE system to Boolean networks [8]. Yet, our study is the first to be dedicated to the *automatic* transformation from a complete quantitative SBML model to Boolean networks. As a complete and automatic process, our pipeline reduces the risk of errors and saves effort and time of biologists. Our results show that SBML2BN succeeds most of the time at recovering small sets of BNs compatible with both the structure and dynamics extracted from the input SBML model.

Overall, SBML2BN is an important building block on which we can build upon. We are investigating strategies to make the pipeline even more efficient, and on more complex models (i.e., for models with more than 10 parents for a component). To go beyond, we plan to take benefit of the *set* of BNs synthesised for a given SBML model by combining and simulating them together, as recently proposed in [6]. We are also investigating how to validate and then aggregate BNs from several SBML models when they concern the same biological system.

³ <https://www.ebi.ac.uk/biomodels/BIOMD0000000044>.

[†] *Availability* All data and programs needed to reproduce the presented results are accessible at <https://gitlab.inria.fr/avaginay/CNA2021>.

Acknowledgements. We thank Hans-Jörg Schurr for his valuable comments and suggestions and Laurine Hubert for helpful comments on an early draft.

References

1. Conda: Anaconda Software Distribution (2021)
2. Aghamiri, S.S., Singh, V., Naldi, A., Helikar, T., Soliman, S., Niarakis, A.: Automated inference of Boolean models from molecular interaction maps using CaSQ. *Bioinformatics* **36**(16), 4473–4482 (2020)
3. Biane, C., Delaplace, F., Melliti, T.: Abductive network action inference for targeted therapy discovery. *Electron. Notes Theor. Comput. Sci.* **335**, 3–25 (2018)
4. Bornstein, B.J., Keating, S.M., Jouraku, A., Hucka, M.: LibSBML: an API library for SBML. *Bioinformatics* **24**(6), 880–881 (2008)
5. Chatain, T., Haar, S., Kolčák, J., Paulev  , L.: Most Permissive Semantics of Boolean Networks. In: Research Report, University Bordeaux, Bordeaux INP, CNRS, LaBRI, UMR5800, F-33400 Talence, France; LSV, ENS Cachan, CNRS, INRIA, Universit   Paris-Saclay, Cachan (France) (2020)
6. Chevalier, S., No  l, V., Calzone, L., Zinovyev, A., Paulev  , L.: Synthesis and simulation of ensembles of Boolean networks for cell fate decision. In: Abate, A., Petrov, T., Wolf, V. (eds.) CMSB 2020. LNCS, vol. 12314, pp. 193–209. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60327-4_11
7. Courtot, M., et al.: Controlled vocabularies and semantics in systems biology. *Molecular Syst. Bio.* **7**(1), 543 (2011)
8. Davidich, M., Bornholdt, S.: The transition from differential equations to boolean networks: a case study in simplifying a regulatory network model. *J. Theor. Bio.* **255**(3), 269–277 (2008)
9. Fages, F., Gay, S., Soliman, S.: Automatic Curation of SBML Models based on their ODE Semantics. Research Report RR-8014, INRIA (2012)
10. Fages, F., Soliman, S.: Abstract interpretation and types for systems biology. *Theoret. Comput. Sci.* **403**(1), 52–70 (2008)
11. Fages, F., Soliman, S.: From reaction models to influence graphs and back: a theorem. In: Fisher, J. (ed.) FMSB 2008. LNCS, vol. 5054, pp. 90–102. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68413-8_7
12. Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: Answer Set Solving in Practice. Morgan and Claypool Publishers, New York (2012)
13. Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., Kummer, U.: COPASI—a complex pathway simulator. *Bioinformatics* **22**(24), 3067–3074 (2006)
14. Kauffman, S.A.: Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Bio.* **22**(3), 437–467 (1969)
15. Keating, S.M., et al.: SBML Level 3 Community members: SBML Level 3: An extensible format for the exchange and reuse of biological models. *Molecular Syst. Bio.* **16**(8), e9110 (2020)
16. Klarner, H., Heinitz, F., Nee, S., Siebert, H.: Basins of attraction, commitment sets, and phenotypes of boolean networks. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **17**(4), 1115–1124 (2020)

17. Klarner, H., Streck, A., Siebert, H.: PyBoolNet: a python package for the generation, analysis and visualization of boolean networks. *Bioinformatics* **33**(5), 770–772 (2016)
18. Lähdesmäki, H., Shmulevich, I., Yli-Harja, O.: On learning gene regulatory networks under the boolean network model. *Mach. Learn.* **52**(1), 147–167 (2003)
19. Liang, S., Fuhrman, S., Somogyi, R.: REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In: Pacific Symposium on Biocomputing, pp. 18–29 (1998)
20. Malik-Sheriff, R.S., et al.: BioModels—15 years of sharing computational models in life science. *Nucleic Acids Res.* **48**(D1), D407–D415 (2020)
21. Mölder, F., et al.: Sustainable data analysis with Snakemake. *F1000Research* **10**, 33 (2021)
22. Ostrowski, M., Paulevé, L., Schaub, T., Siegel, A., Guziolowski, C.: Boolean network identification from perturbation time series data combining dynamics abstraction and logic programming. *Biosystems* **149**, 139–153 (2016)
23. Paulevé, L., Kolčák, J., Chatain, T., Haar, S.: Reconciling qualitative, abstract, and scalable modeling of biological networks. *Nat. Commun.* **11**(1), 4256 (2020)
24. Schwab, J.D., Kühlwein, S.D., Ikonomi, N., Kühl, M., Kestler, H.A.: Concepts in Boolean network modeling: what do they all mean? *Comput. Struct. Biotechnol. J.* **18**, 571–582 (2020)
25. Thomas, R.: Boolean formalization of genetic control circuits. *J. Theor. Biol.* **42**(3), 563–585 (1973)
26. Vaginay, A., Boukhobza, T., Smail-Tabbone, M.: Automatic Synthesis of Boolean Networks from Biological Knowledge and Data. In: Dorronsoro, B., Amodeo, L., Pavone, M., Ruiz, P. (eds.) OLA 2021. CCIS, vol. 1443, pp. 156–170. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85672-4_12



Can Dynamic Functional Connectivity Be Used to Distinguish Between Resting-State and Motor Imagery in EEG-BCIs?

Paula G. Rodrigues^{1,2}(✉), Carlos A. Stefano Filho^{2,3}, André K. Takahata^{1,2}, Ricardo Suyama^{1,2}, Romis Attux^{2,4}, Gabriela Castellano^{2,3}, João R. Sato⁵, Sławomir J. Nasuto⁶, and Diogo C. Soriano^{1,2}

¹ Center for Engineering, Modeling and Applied Social Sciences - Federal University of ABC (UFABC), São Bernardo do Campo, SP, Brazil

² Brazilian Institute of Neuroscience and Neurotechnology (BRAINN), Campinas, Brazil

³ Neurophysics Group, “Gleb Wataghin” Institute of Physics, University of Campinas (UNICAMP), Campinas, SP, Brazil

⁴ Department of Computer Engineering and Industrial Automation (DCA), UNICAMP, Campinas, Brazil

⁵ Center for Mathematics, Computing and Cognition, UFABC, Santo André, Brazil

⁶ Brain Embodiment Laboratory, Biomedical Engineering, School of Biological Sciences, University of Reading, Reading, UK

Abstract. Graph theory has been widely and efficiently applied to characterize brain functioning, ranging from the diagnosis of pathologies (e.g. depression, Alzheimer, schizophrenia, etc.) to investigations of cognitive processes in neuroscience. Recently, the use of graph-based strategies through functional connectivity (FC) analysis has shown to be an interesting option for feature extraction in motor imagery brain-computer interfaces (MI-BCIs) - an alternative communication system that does not require the use of classical biological efferent pathways, mapping brain signals directly to control external assistive devices. Although FC has been used in such context, the dynamics of FC under motor imagery has rarely been taken into account, which outlines an essential requirement for online BCI operation. Therefore, this study aims to evaluate the applicability of dynamic functional connectivity (dFC) to differentiate resting-state and motor imagery in electroencephalography (EEG)-based BCI. We evaluated the classification performance of classical markers, as defined by event-related desynchronization, static FC and the dynamic FC scenario. The analysis includes two different similarity criteria for estimating the FC matrix and four different graph-based metrics applied to a representative EEG dataset with 35 subjects. The obtained results point to the potential and accompanying challenges of using dFC in the context of MI-BCIs, in addition to providing insights concerning the motor network activity organization during motor imagery.

Keywords: Brain-computer interface · Dynamic functional connectivity · Electroencephalography · Motor imagery · Time-varying graphs

1 Introduction

Motor imagery (MI) can be defined as a mental simulation of movements of the body (e.g. the movement of a hand) without the actual muscular activity and has an important role in motor learning and control [1]. Sharing some similarity with the brain activity during motor execution, different brain regions are typically involved in the MI process, such as the supplementary motor and premotor areas. This makes the use of MI an appealing approach to study the neural control of body movements [2], and widely used for the control of brain-computer interfaces.

A brain-computer interface (BCI) is a system that can provide an alternative communication pathway, mapping the brain activity directly into commands for assistive devices by the induction of brain patterns associated with specific activities or stimuli representing the desired outcomes according to a specific BCI paradigm protocol [3]. In the case of MI-BCIs, the system is actively controlled by the imagination of different body movements, with the left and right-hand MI being the most commonly employed strategies. From the neurophysiological perspective, motor activity evokes a task-dependent synchronization or desynchronization of neuronal population, leading to the stereotypical modulation of the power of the electroencephalography (EEG) rhythms, usually on the alpha (or mu) and beta bands. The synchronization and desynchronization of the cortical neurons due to a motor activity are known as event-related synchronization (ERS) and event-related desynchronization (ERD), respectively, and they represent important biomarkers for the study of MI activities [4].

Besides classical event-related bandpower fluctuations, brain organization has also been characterized by functional connectivity (FC) evaluation, defined as a measure of inter-region statistical similarity [5]. The FC characterization is usually summarized through graph-based metrics and has been widely applied to investigate a variety of brain conditions/diseases, such as Alzheimer's disease [6] and psychiatric disorders [7]. Using this theoretical approach, it is possible to transform different brain imaging recordings into networks representing the FC pattern of a task or disease. FC has been used to gain a deeper knowledge of the brain organization during MI activities, and the patterns associated to each task have been studied in the context of BCI as an alternative to the traditional EEG biomarkers [8], such as to obtain more robust BCI features for the challenging problem of short-time EEG classification.

Different approaches have been explored to extract the FC information of EEG signals. In [9], when investigating the importance of the fronto-parietal attention network in the MI-BCI performance, the authors analyzed structural and functional connectivity features and observed that they were significantly associated with performance depending on the brain region considered. The researchers noticed that the eigenvector centrality and cortical thickness of the left inferior

parietal lobe could identify low-aptitude BCI users with high accuracy (83%). In [8], left and right-hand FC was characterized using motifs and five graph metrics (strength, clustering coefficient, characteristic path length, betweenness centrality, eigenvector centrality). They observed that the selection of electrodes increased the classification performance compared to using all electrodes available, and the results of the beta band were better than for the mu band. In general, the FC features individually did not outperform the classical approach (given by power spectral density), but when the graph metrics were combined, similar performances with a lower number of features could be attained [8].

Still considering the FC analysis for BCI systems, in [10], correntropy was compared with Pearson and Spearman similarities, with FC characterized using degree, betweenness and eigenvector centralities. The authors observed that the correntropy defined a promising similarity option for discriminating the MI task. In [11], on the other hand, focusing on comparing different similarity measures using four graph metrics, it was shown that a recurrence-based similarity measure was significantly better than the tested ones (mean phase coherence, Pearson's and Spearman's correlation), and the eigenvector centrality was identified as an interesting graph-based feature for online processing. Finally, a recent study investigated the neurophysiological factors that could determine the BCI performance by means of the FC analysis, since a significant amount of users are not able to control MI-BCIs using the classical features. Applying the imaginary part of coherency, the study showed that the BCI performance is dependent on the amplitude of sensorimotor oscillations (ERD/S) and that it is related to the sensorimotor connectivity measured in the training session [12].

Although informative, these works usually consider the MI period as a whole for evaluating FC, estimating a single connectivity matrix, defining a “static” picture of the functional organization (the static FC). However, since the MI phenomenon involves the interaction of different brain regions over time [2], an analysis considering the dynamic aspects of FC during the MI period - the dynamic functional connectivity (dFC) - could reveal more insights about the brain activity underpinning different stages of the MI and also contrasting different imagery tasks. This kind of analysis is not as common as the static approach due to its challenges (e.g. dealing with smaller time windows at a higher computational cost), but it has been able to extract relevant information concerning the differentiation of BCI tasks [13, 14], defining, in a wider sense, a general brain processing paradigm. Moreover, it is important to emphasize that the BCI is *per se* a dynamic classification task, requiring buffer manipulation and online processing, which better maps onto the dFC scenario.

Having these issues in mind, this work presents a comparison between different feature extraction aiming to discriminate rest and MI conditions. The comparison takes into account the classical ERD and its dynamics, as well as the evaluation of static FC and dynamic FC using two different similarity criteria: phase coherence through the phase lag index, and rhythm correlation through Pearson correlation. Four different graph metrics defined over a weighted graph were used to summarize the connectivity topology. The analysis was performed

considering 35 subjects from a representative public dataset [15] and results in essential insights into the use and performance of dFC for MI-BCIs.

2 Materials and Methods

2.1 Dataset

A representative and publicly available BCI MI dataset comprising recordings from 50 right-handed subjects for two MI tasks (right-hand and left-hand sequential finger tap) [15] was used. The data were collected using 64 Ag/AgCl active electrodes, at a sampling rate 512 Hz. The experimental procedure consisted of a black screen with a fixation cross lasting 2 s to draw the subject's attention, followed by 3 s of the imagery task and 2 s of rest, giving a total of 7 s for each trial. For both tasks, 100 or 120 trials were collected. The first 100 trials for each imagery task were used aiming uniformity. The trials pointed out as bad (due to artifacts or high correlation with electromyography) were excluded from the analysis. A detailed data description can be found in [15].

2.2 Preprocessing

To eliminate the noise of the EEG signals and enhance the relevant information, a bandpass filter between 1-55 Hz (6th order) and an IIR notch filter 60 Hz were applied. Eye blinks were automatically removed by linear regression based on the blinks recorded at FP1 and FP2 electrodes. To minimize volume-conduction effects and other artifacts, a Laplacian spatial filter was applied [16].

2.3 ERD/S Computation

To check the presence of a latent MI activity before FC evaluation, the ERD or ERS (expressed here as ERD/S for conciseness, since it represents the same quantity) was estimated and analyzed. The ERD/S is classically defined as the percentage of the relative difference between the signal power in a given instant ($P(t)$) and its baseline (B) – interval before the cue – in a chosen frequency band [4], as shown in Eq. 1.

$$ERD/S = \frac{P(t) - B}{B} \times 100 \quad (1)$$

The signal power was evaluated using the magnitude of the Hilbert transform after bandpass filtering (4th order Butterworth - 8 to 16 Hz) to yield the mu rhythm. However, in contrast to the classical definition, in which the baseline is estimated after the inter-trials average for ensemble analysis, the ERD/S was computed here on the individual trials to meet the BCI classification requirements. The baseline interval of this study was set between -1.5 s and 0 s.

2.4 Exclusion Criteria

As some subjects are not able to perform MI tasks (around 54% according to a recent study) [17], our study considered only subjects with some latent response in an attempt to avoid problems resulting from a high degree of BCI illiteracy. The following criteria were used to exclude the subjects from the final analysis, leading to a final sample of 35 subjects:

1. Subjects with strong artifacts in the mean single-trial ERD/S. After visual inspection, eight subjects were excluded.
2. Subjects with more than 70% of the trials correlated with the electromyography and/or with bad EEG amplitudes. Two subjects were excluded based on this criteria.
3. Subjects with no statistically significant difference between mean resting state and mean MI intervals in single-trial ERD/S. Four electrodes in the motor region were chosen (C3, C4, CP3 and CP4) and a permutation test was performed considering the temporal averaging value of the ERD/S in the rest interval and MI period (between 1 and 3 s, to exclude the transition). The p-value was estimated after 5000 runs and the significance level was set to 0.05. If the signal from at least two electrodes were significantly different concerning left-hand trials and two for the right-hand trials, the subject was considered as having the minimum MI ability and he/she was included in the studied population. After this selection, five subjects were excluded.

2.5 Functional Connectivity Evaluation

Two different similarity measures were used to evaluate FC in the sensor space (EEG electrodes, defining the brain network nodes): Pearson's correlation in the EEG's mu rhythm and the phase-lag index in this same rhythm. Pearson's correlation is a time-domain measure used to detect linear dependencies between two time series. It is bounded between -1 and 1 , with 0 indicating absence of similarity. It can be defined by Eq. 2, in which $cov(i, j)$ denotes the covariance between signals i and j , and var is the signals' variance [18].

$$P_{i,j} = \frac{cov(i, j)}{\sqrt{var(i)var(j)}} \quad (2)$$

The phase-lag index (PLI) considers the distribution of phase angle differences toward positive or negative sides of the imaginary axis on a complex plane. It is computed by taking the mean value of the sign of the imaginary part of the cross-spectral density, as shown in Eq. 3, in which $img(S)$ denotes the imaginary part of the cross-spectral density at time t and sgn is the sign of a number (-1 for negative values, 1 for positive values, 0 if zero) [16]. In the case of this measure, spurious connectivity returns phase angle differences distributed around zero radians and non spurious connectivity gives a distribution concentrated predominantly on the positive or negative sides of the imaginary axis [16].

$$PLI_{i,j} = \left| \frac{1}{n} \sum_{t=1}^n sgn(img(S_{ijt})) \right| \quad (3)$$

After the computation of the pairwise similarity, the topology of the functional network was characterized at each network node using four graph metrics: clustering coefficient (CC), strength (S), closeness centrality (CL) and eigenvector centrality (EC). The graph metrics were estimated considering the absolute value of the connectivity matrices between each pair of electrodes leading to a weighted undirected graph. No binarization was applied, in order to avoid empirical threshold choices.

The standard node clustering coefficient characterizes the network segregation and measures the average probability that two neighbours of a node are also interconnected, thus, estimating the density of triangles in the network. In the case of weighted connectivity matrices, the intensity of the subgraph triplets is given by the geometric mean of its edge weights, as defined in Eq. 4, in which k_i is the degree of the node i [19, 20]. The weights were scaled in relation to the maximum edge weight in the graph, $\hat{w}_{ij} = w_{ij}/max(w)$.

$$CC_i = \frac{2}{k_i(k_i - 1)} \sum_{j,h} (\hat{w}_{ij}\hat{w}_{ih}\hat{w}_{jh})^{\frac{1}{3}} \quad (4)$$

Node strength is defined as the sum of the strengths of all the node connections, providing a measure of node importance, since strongly-connected elements indicate potential hubs [21]. The metric is defined in Eq. 5, in which w_{ij} denotes the weight of the connection between a node i and j .

$$S_i = \sum_{j \neq i} w_{ij} \quad (5)$$

The eigenvector centrality takes into account the connection of a node and its neighbors, leading to a measure of importance of a node. By doing so, nodes that are connected to less influential ones have a lower EC value compared to nodes connected to more influential ones. To estimate the EC value, it is necessary to compute the eigenvectors x and eigenvalues λ of the adjacency matrix A between the nodes i and j , as expressed in Eq. 6 [21].

$$EC_i = x_i = \frac{1}{\lambda} \sum_{j=1}^N A_{ij}x_j \quad (6)$$

Lastly, the closeness centrality of a node is a metric which takes into account the distance between nodes (here defined as $-\log(w_{ij})$), with its value based on the inverse of the sum of all shortest paths that a node i can reach (C_i). This metric aims to detect nodes that can spread information efficiently in a network, and can be defined by Eq. 7, in which R_i is the number of reachable nodes from node i and N the number of nodes in the graph [19, 21].

$$CL_i = \left(\frac{R_i}{N - 1} \right)^2 \frac{1}{C_i} \quad (7)$$

2.6 Static and Dynamic Functional Connectivity

The graph-based metrics were used as features for a classifier considering the static and dynamical FC scenarios. All the classifiers were obtained using linear discriminant analysis. In the static case, the resting state condition was defined based on the baseline interval (between -1.75 s and 0 s) and MI class from 0 s to 3 s (acquisition protocol). A connectivity matrix was computed for each trial and the four mentioned graph metrics were estimated to characterize the networks. The classification was performed using the hold-out method (30 times), with 75% of the trials being used to train the classifier. The cross-validation was performed within-subjects, without mixing data of different subjects. Each electrode was classified individually, without any combination of attributes, as a way to determine if there was any electrode with higher performance than the others.

In the dFC scenario, the whole EEG signal (7 s) was used in the connectivity analysis. To segment the signals and estimate the dFC, the sliding window size used was set to 256 samples (0.5 s) and the sliding window step was set to 102 samples (approx. 0.2 s). The same number of windows of rest and MI was used to train the classifier. Windows with a greater number of samples related to rest/MI were considered part of the rest/MI class, respectively. To avoid transitions between mental states, windows between -1.5 s and 0 s (region of 1.5 s duration) were defined as the training set related to the rest class and windows between 1.5 s and 3 s (1.5 s duration) were defined as part of the MI class.

Given the fact that the position of the time windows is important for the study of the dFC behavior, a hold-out method was used to split the dataset considering not the windows, but the trials. By doing so, 75% of the trials were used to train the classifier. Hold-out was performed 30 times, classifying all the windows in the dataset. Each electrode was classified individually, similarly to the static approach.

In the present study, because the aim was to find differences between rest and MI, the right-hand and left-hand task were concatenated in a ‘motor imagery’ class and differences between these two classes were not characterized.

3 Results

3.1 ERD/S and Static FC Classification Performances

Aiming to illustrate the classical power-based response evoked during MI, thus attesting the presence of the MI phenomenon, Fig. 1 shows the mean single-trial ERD/S and the standard deviation considering the subjects’ mean ERD/S ($N = 35$). As expected, a clear desynchronization (power decrease) in the motor imagery interval in the electrodes associated to the motor cortex (e.g. C3 and C4) can be observed, with a higher contralateral EEG desynchronization (greater power decrease) respective to the movement imagined (higher ERD at C3 electrode under right-hand task and at C4 under left-hand task).

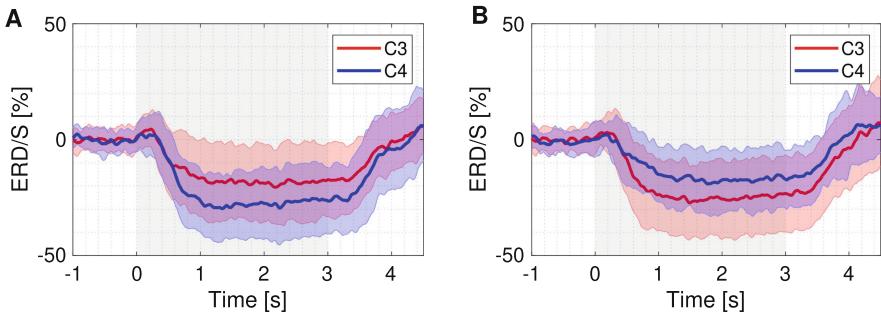


Fig. 1. Mean single-trial ERD/S of C3 and C4 for (A) left-hand and (B) right-hand trials considering the investigated population ($N = 35$). The shaded area represents the standard deviation and the area in gray indicates the MI region.

Clearly, after the cue, there is a delay for the beginning of the desynchronization, which takes almost 1 s to stabilize. Moreover, after the end of the motor imagery ($t = 3$ s), a slower recovery takes place, during almost 0.5 s before starting the synchronization phenomenon towards the resting state. Thus, there is a clear phase transition from rest to MI, with a stable dynamics during the respective phases.

Figure 2 shows the mean classification accuracy of all EEG electrodes considering the static FC estimated through the different similarity criteria, and four graph-based metrics and also the performance obtained by the ERD/S marker.

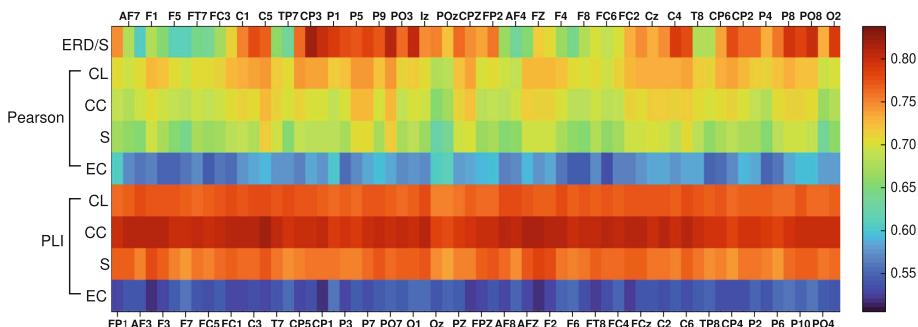


Fig. 2. Mean classification accuracy (rest vs. MI) of the electrodes for the static connectivity and mean ERD/S features for the population ($N = 35$).

For the ERD/S marker, the classification performance significantly varied depending on electrode position, while the static FC tends to provide a more homogeneous performance. In general, the static FC accuracy depends more on similarity measure than on the graph-based metric. For both similarity approaches, the metric with the worst performance was the eigenvector centrality, while there was not a single best metric. Table 1 shows the peak performance,

considering all EEG electrodes, in which we see that the combination PLI + CC obtained the highest performance.

Table 1. Mean peak accuracy for the two similarity measures analyzed for the clustering coefficient (CC), eigenvector centrality (EC), closeness centrality (CL) and strength (S). Mean \pm standard deviation (std).

	CC	EC	CL	S
Pearson	0.79 \pm 0.06	0.73 \pm 0.06	0.80 \pm 0.06	0.79 \pm 0.06
Phase-lag index	0.86 \pm 0.04	0.65 \pm 0.06	0.82 \pm 0.03	0.83 \pm 0.04

The mean performance was computed considering the peak performance for each subject, which implies that the chosen electrodes may be different for each subject. For the ERD/S the peak accuracy was 0.89 ± 0.06 .

3.2 Dynamic ERD/S and Functional Connectivity Classification Performance

Figure 3 shows the mean population classification accuracy over time using the ERD/S estimated over sliding windows of 0.5 s with an overlap of 102 samples (approx. 0.2 s). The line at the top of the electrode values refers to the position of the windows used to train the classifier.

It is possible to observe that the transitions between rest to MI and vice-versa are characterized by the abrupt change of the actual label, which is not immediately accompanied by the classifier prediction. Thus, given the latency of the electrophysiological response, a region of misclassified windows can be observed (blue windows). After such transitions, a representative classification peak accuracy is achieved (0.90 ± 0.07 - mean population accuracy \pm std).

From the dFC standpoint, Fig. 4 shows the mean accuracy for Pearson's correlation and the four graph-based metrics. For the sake of visualization, only the 20 best electrodes were represented. In general, the dFC could not correctly classify the rest interval, but was able to achieve reasonable classification performances during the MI interval, specially for the closeness centrality metric. On the other hand, eigenvector centrality was the metric with worst differentiation between rest and MI. The results obtained using PLI are analogous to Pearson's, and the figure was omitted for conciseness.

To summarize dFC results, Table 2 presents the mean peak performance considering all EEG electrodes and windows. The mean performance was computed considering the peak performance for each subject, which implies that the chosen electrodes and windows may be different for each subject in the analysis.

Table 2 shows that the Pearson provided the best performance, with the closeness centrality being the metric achieving the best results. Comparing them to the traditional approach (ERD/S), the Pearson's correlation was significantly different for almost all graph metrics (except for CC), showing that only

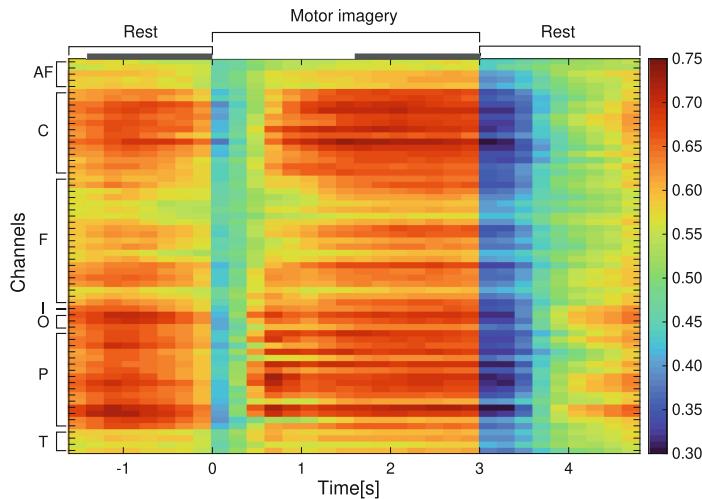


Fig. 3. Mean accuracy of the population for each window considering the classification between rest and MI for the ERD/S. The defined labels for each signal region are indicated in the picture. The letter on the side of the figure is related to the electrodes: C for central and centro-parietal electrodes; AF for AF electrodes; F for frontal, fronto-parietal, fronto-central and fronto-temporal electrodes; I for Iz electrode; O for occipital electrodes; P for parietal and parieto-occipital electrodes; and T for temporal electrodes

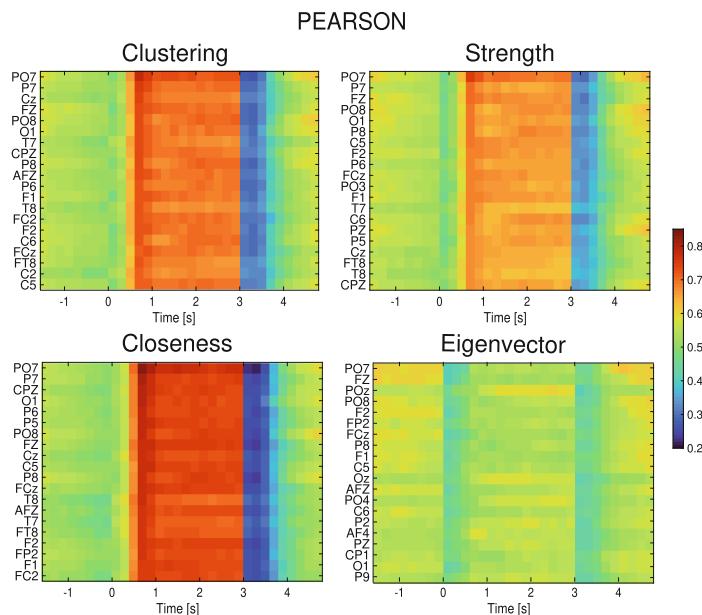


Fig. 4. Mean accuracy over time for the 20 best electrodes using Pearson's correlation

Table 2. Mean peak accuracy for the methods analyzed for the clustering coefficient (CC), eigenvector centrality (EC), closeness centrality (CL) and strength (S).

	CC	EC	CL	S
Pearson	0.88 ± 0.09	0.79 ± 0.09	0.92 ± 0.08	0.86 ± 0.09
Phase-lag index	0.85 ± 0.10	0.72 ± 0.07	0.90 ± 0.08	0.83 ± 0.09

the closeness centrality performance overcame the ERD/S (p-value < 0.0001, Kruskall-Wallis with Dunn's multiple comparisons test correction). Among the graph metrics for Pearson, all graph metrics were significantly different from each other (p-value < 0.0001, with the exception of CC vs S, p-value = 0.015).

4 Discussion and Conclusions

Concerning our results, it was possible to observe that the graph metrics were able to discriminate between both states, and their success was related to the graph metric applied. The difference in accuracy between the peak static FC and dFC was small, with higher accuracy for the dFC for some combinations. This finding indicates that not many samples are needed for a satisfactory classification since the difference between these mental states is substantial, which may support the use of this feature in online scenarios. However, the brain state related to the time window was important for the classification given that the windows related to the MI obtained a higher performance than the resting-state ones for the dFC approach. The metric with the highest performance changed between static and dFC approaches, but the metric with the worst performance was the EC, differently from the finding of [8,9,11]. This may be due to the fact that we used a fully connected matrix to characterize the connectivity and this choice could result in lower performance for this metric. Moreover, the combination of features was not performed in this study, which could increase the accuracy, as observed in [8]. Besides that, it was possible to verify that the transition between mental states is not abrupt, causing a period of confusion in terms of accuracy. The dFC classification using Pearson and CL achieved better performance than the classical biomarker (ERD/S), showing the feasibility of the dFC to differentiate resting-state vs MI in MI-BCI, but a comparison to BCI state-of-the-art techniques, such as multi-band Common Spatial Patterns, is out of the scope of this study and may be addressed in future investigations.

Acknowledgments. PGR thanks CAPES financial support; GC thanks FAPESP (n. 2013/07559-3); DCS thanks FAPESP (n. 2019/09512-0) and CNPq (n. 306298/2020-1); JRS thanks FAPESP (n. 2018/21934-5).

References

1. Diez, P., et al.: Smart Wheelchairs and Brain-Computer Interfaces. Elsevier, Amsterdam (2018)

2. Hanakawa, T.: Organizing motor imageries. *Neurosci. Res.* **104**, 56–63 (2016)
3. Wolpaw, J., Wolpaw, E.W.: *Brain-Computer Interfaces: Principles and Practice*. Oxford University Press, Oxford (2012)
4. Pfurtscheller, G., Lopes Da Silva, F.H.: Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clinic. Neurophys.* **110**(11) 1842–1857 (1999)
5. Sporns, O.: *Networks of the Brain*. The MIT Press, Cambridge (2011)
6. Babiloni, C., Del Percio, C., Lizio, R., Noce, G., Lopez, S., Soricelli, A., et al.: Abnormalities of resting-state functional cortical connectivity in patients with dementia due to Alzheimer's and Lewy body diseases: an EEG study. *Neurobiol. Aging* **65**, 18–40 (2018)
7. Zhang, Y., et al.: Identification of psychiatric disorder subtypes from functional connectivity patterns in resting-state electroencephalography. *Nat. Biomed. Eng.* **5**(4), 309–323 (2020)
8. Stefano Filho, C.A., Attux, R., Castellano, G.: Can graph metrics be used for EEG-BCIs based on hand motor imagery? *Biomed. Sig. Process. Control* **40**, 359–365 (2018)
9. Zhang, T., Liu, T., Li, F., Li, M., Liu, D., Zhang, R., et al.: Structural and functional correlates of motor imagery BCI performance: insights from the patterns of fronto-parietal attention network. *NeuroImage* **134**, 475–485 (2016)
10. Uribe, L.F.S., et al.: A correntropy-based classifier for motor imagery brain-computer interfaces. *Biomed. Phys. Eng. Exp.* **5**(6), 065026 (2019)
11. Rodrigues, P.G., Filho, C.A.S., Attux, R., Castellano, G., Soriano, D.C.: Space-time recurrences for functional connectivity evaluation and feature extraction in motor imagery brain-computer interfaces. *Med. Biologic Eng. Comput.* **57**(8), 1709–1725 (2019). <https://doi.org/10.1007/s11517-019-01989-w>
12. Vidaurre, C., Haufe, S., Jorajuria, T., Müller, K.R., Nikulin, V.V.: Sensorimotor functional connectivity: a neurophysiological factor related to BCI performance. *Front. Neurosci.* **14** 1278 (2020)
13. Daly, I., Nasuto, S.J., Warwick, K.: Brain computer interface control via functional connectivity dynamics. *Pattern Recogn.* **45**(6), 2123–2136 (2012)
14. Shamsi, F., Haddad, A., Najafizadeh, L.: Early classification of motor tasks using dynamic functional connectivity graphs from EEG. *J. Neural Eng.* **18**(1), 016015 (2020)
15. Cho, H., Ahn, M., Ahn, S., Kwon, M., Jun, S.C.: EEG datasets for motor imagery brain-computer interface. *GigaScience* **6**(7), 1–8 (2017)
16. Cohen, M.X.: *Analyzing Neural Time Series Data: Theory and Practice*, vol. 53, 1st edn. The MIT Press, Cambridge (2014)
17. Lee, M.H., Kwon, O.Y., Kim, Y.J., Kim, H.K., Lee, Y.E., Williamson, J., et al.: EEG dataset and OpenBMI toolbox for three BCI paradigms: an investigation into BCI illiteracy. *GigaScience* **8**(5), 1–16 (2019)
18. Jalili, M.: Functional brain networks: does the choice of dependency estimator and binarization method matter? *Scientif. Rep.* **6**(1), 29780 (2016)
19. Fornito, A., Zalesky, A., Bullmore, E.T. (eds.): *Fundamentals of Brain Network Analysis*, 1st edn. Elsevier, Amsterdam (2016)
20. Onnela, J.P., Saramäki, J., Kertész, J., Kaski, K.: Intensity and coherence of motifs in weighted complex networks. *Phys. Rev. E - Stat., Nonlinear, Soft Matter Phys.* **71**(6), 1–4 (2005)
21. Newman, M.: *Networks: An Introduction*. Oxford University Press, Oxford (2010)

Network Medicine



Inferred Networks and the Social Determinants of Health

Prashant Sanjel and John Matta^(✉)

Southern Illinois University Edwardsville, Edwardsville, IL 62026, USA
{psanjel,jmatta}@siue.edu

Abstract. This paper explores the social determinants of health through a network science based approach to analyzing the Latino MSM Community Involvement (LMSM-CI) dataset. Data are clustered to determine identifying characteristics of groups of participants in 3 categories: high self esteem, susceptibility to alcohol abuse, and HIV positive status. A question arises as to the best methodology for inferring a graph from the data, as well as for clustering and analyzing the network. To that end we use 4 different graph inference methods: inverse covariance selection (Glasso), neighborhood selection (MB), Sparse Correlations for Compositional data (SparCC) and the traditional k-Nearest Neighbors (kNN). For each inference we test 4 different clustering methods: Louvain, Leiden, NBR-Clust with VAT, and NBR-Clust with integrity. Surprisingly, the Glasso and MB inference methods produce better clusterings than kNN, as determined by a suite of internal evaluation measures. The most promising clusterings are visualized and their properties are analyzed.

Keywords: Graph inference · Clustering · Machine learning · Health data analysis

1 Introduction

The desire to find a happy and healthy life has led to the emergence of Social Determinants of Health (SDoH). The Centers for Disease Control (CDC) outlines SDoH as “conditions in the environments where people are born, live, learn, work, play, worship, and age that affect a wide range of health, functioning, and quality-of-life outcomes and risks” [11]. Those conditions can be a catalyst for better health and prosperity. Understanding data on SDoH such as education, income, employment, age, and more can aid in attempts to improve community health. The use of machine learning algorithms such as clustering on graphs inferred from health and medical datasets presents opportunities towards personalized medicine, and a healthier and better quality of life.

There are benefits to inferring a network from existing data and working with it in network form [8]. These include the ability to simplify data to a nodes-and-edges representation, the opportunity to employ a wide range of existing algorithms and tools, the ability to learn from effective visualization of the information, and the ability to observe the dynamic behaviors of individuals while considering various factors or attributes that may influence each other [7].

Due to technical advances in genetic sequencing technologies there has been a large influx of new data. Desire to exploit this data has lead to a variety of graph inference techniques, including those examined here: inverse covariance selection (Glasso) [15], Meinshausen and Bühlmann (MB) neighborhood selection [23], Sparse Correlations for Compositional data (SparCC) [14], and the traditional k-Nearest Neighbors (kNN). Due to expanded use of electronic medical records, there is a large amount of newly-available medical data. This paper seeks to determine the applicability of these graph inference methods to applications with health and medical data. Surprisingly, Glasso and MB give strong results, which is a contrast with previous papers that preferred kNN graphs [18].

Social determinant data used here are from Jesus Ramirez-Valles' *Latino MSM Community Involvement: HIV Protective Effects* (LMSM-CI) survey [27], the original purpose of which was to study HIV in male Latino populations. We examine social determinants leading to target variables of high self esteem, alcohol abuse risk, and positive HIV status via graph inference methods and clustering on the LMSM-CI dataset. The data are available online through application with the National Addiction & HIV Data Archive Program¹. This research was approved by the SIUE Institutional Review Board.

2 Related Work

In this study we have inferred Glasso, MB, SparCC, and kNN graphs. The Glasso method is based on graphical lasso regression, which has been widely used in diverse applications like financial interpretation [4] and facial recognition [32]. KNN is a popular network inference method, with previous medical data applications including predicting sleep disorders [26] and analyzing stroke biomarkers [12]. In [19] the graph methods that we see in this paper are tested with microbial associations. They have also have been applied to other domains, such as metagenomics and inflammatory bowel disease [1], ribosomal and micro RNA [14], and spoilage of food [25].

Machine learning approaches have been employed in the study of social determinants of health [17], such as in [28], where social and economic factors are used to predict medical data like blood pressure and body mass index. In other examples, social and economic factors have been used to study the causes of addiction [2] and to evaluate mental disorders [3].

This paper focuses on four network-based methods of community detection: Louvain, Leiden, and NBR-Clust with VAT and integrity. The best clusterings are chosen by cluster evaluation methods, in accordance with work previously done in [18]. Machine learning tools like clustering have been used on graphs inferred from medical data to analyze and understand conditions such as autism spectrum disorder [22]. Clustering has been used to predict drug-drug reactions [30] and analyze DNA sequences [16]. In this study we rigorously analyze graph inference and machine learning techniques as applied to the LMSM-CI dataset.

¹ <https://www.icpsr.umich.edu/web/pages/NAHDAP/index.html>.

3 Methods

3.1 The LMSM-CI Dataset

The data used in this study are from the Latino MSM Community Involvement: HIV Protective Effects survey (LMSM-CI) [27]. The data were collected in 2003–2004 and consist of 323 samples of Latino men who have sex with men (MSM) in the Chicago, Illinois, USA metro area and 320 samples in the San Francisco, California, USA area. The study consists of over 900 variables, with many variables related to the social determinants of HIV in the Latino MSM community. The foremost aim of the survey is to determine if Latino MSMs are more likely to take precautions against HIV if they are involved in community activities. The large amount of data collected makes this dataset relevant not just to HIV, but to other determinants as well.

3.2 Data Curation

Many of the variables contained in the LMSM-CI dataset were meta-data, such as the IDs of the recruiter and subsequent recruits of the sample. Also, the survey contained many questions involving participation in specific charities or social groups. To simplify analysis, these variables were removed. The remaining multi-valued variables were converted to binary variables using one-hot encoding. As an example of one-hot encoding, a variable such as *employment-status* with possible answers *employed*, *unemployed*, and *on-disability*, is changed to three variables with yes/no answers: *employment-status:employed*, *employment-status:unemployed*, and *employment-status:on-disability*.

The National Institute on Alcohol Abuse and Alcoholism (NIAAA) [24] defines heavy alcohol abuse risk for men as binge drinking for more than 4 days in the past month, where binge drinking is defined by the Substance Abuse and Mental Health Services Administration (SAMHSA) as drinking more than 5 alcoholic drinks. We created the target variable *Alcohol Risk* based on these definitions and the LMSM-CI variables concerning frequency of alcohol consumption and number of drinks on a typical day. The self esteem target we study represents an affirmative answer in the survey to “I have a positive attitude about myself.”

Feature selection was performed as described in [12], with stratified testing and training sets. The step forward algorithm was used to select the features giving the best performance as determined by logistic regression. The end result was a list of the most important 70, 20 and 15 variables for each of the 3 target variables being studied.

3.3 Graph Inference

The LMSM-CI data were converted into graph format using four graph inference methods. These inference methods assume sparse networks, so in all cases

parameter settings were adjusted to produce a connected graph with the minimum number of edges. The first graph inference method is the widely used k-Nearest Neighbors (kNN), where distances are calculated between each pair of proband vectors \vec{uv} , and an edge is placed between node u and its k shortest distance neighbors v . The kNN graphs were created using the CCCD R package.

The second graph inference method is Glasso. This is a fast method for “estimating sparse graphs by a lasso penalty applied to the inverse covariance matrix” [15]. The third inference method, Meinshausen and Bühlmann (MB) [23], uses neighborhood selection. It “estimates a sparse graphical model by fitting a lasso model to each variable, using the others as predictors” [15]. The last method is Sparse Correlations for Compositional data (SparCC) [14]. SparCC uses linear Pearson correlations to infer a network of associations. This approach has been shown to ameliorate unreliable results that can occur with correlation analysis methods. Glasso, MB, and SparCC networks were created using Sparse Inverse Covariance Estimation for Ecological Association Inference (SPIEC-EASI) [19].

3.4 Clustering

Four clustering methods were used. Louvain [6] is a popular, low-time-complexity algorithm that seeks to produce clusters that maximize modularity, a well-known method of quantifying the goodness of a clustering, as measured against a random clustering. The Leiden algorithm is meant to represent an improvement over Louvain. It “converges to a partition in which all subsets of all communities are locally optimally assigned. The Leiden algorithm is faster than the Louvain algorithm and uncovers better partitions. In addition, it has been proved that the Leiden algorithm yields communities that are guaranteed to be connected” [29]. The NBR-Clust framework [21] uses network resilience measures to partition a graph into clusters. It identifies an attack set of nodes $S \in V$, whose removal partitions the network into some number of disconnected components. Resilience measures used with NBR-Clust are integrity, which is defined as

$$I(G) = \min_{S \subset V} \{|S| + C_{max}(V - S)\}, \quad (1)$$

and vertex attack tolerance (VAT), which is defined as

$$VAT(G) = \min_{S \subset V} \left\{ \frac{|S|}{|V - S - C_{max}(V - S)| + 1} \right\}, \quad (2)$$

where V is the set of vertices, S is the attack set, and C_{max} is the size of the remaining largest connected component. Resilience measures are approximated using betweenness centrality [20]. For all three clustering methods, the number of clusters is not specified *a priori*.

3.5 Internal Evaluation Measures

We quantified the success of clustering results with 5 internal evaluation measures, using the methodology presented in [18, 22]. Cluster evaluation measures

Table 1. Cluster evaluation results for self esteem

Cluster Algorithm	#	Glasso					SparCC				
		Davies-Bouldin	Silhouette	Calinski-Harabasz	Baker-Hubert	Hubert-Levine	Davies-Bouldin	Silhouette	Calinski-Harabasz	Baker-Hubert	Hubert-Levine
Louvain	15	1.911	0.168	4.376	0.546	0.248	2.216	-0.021	1.589	0.240	0.396
	20	1.558	0.199	5.593	0.708	0.155	2.399	0.033	1.834	0.223	0.403
	70	2.354	0.054	2.293	0.426	0.330	2.275	2.275	1.841	0.297	0.384
Leiden	15	2.402	0.174	4.416	0.650	0.209	2.137	0.031	2.014	0.262	0.369
	20	1.558	0.199	5.593	0.708	0.155	2.387	0.028	1.820	0.215	0.402
	70	2.648	0.093	2.648	0.526	0.281	2.426	0.042	2.038	0.321	0.372
VAT	15	1.974	0.096	3.816	0.573	0.239	1.711	-0.084	1.231	0.008	0.499
	20	2.130	0.075	4.907	0.458	0.262	1.718	-0.210	0.981	-0.105	0.563
	70	2.351	-0.164	1.511	0.180	0.429	1.243	-0.226	0.822	-0.182	0.550
Integrity	15	1.974	0.096	3.816	0.573	0.239	1.794	-0.131	1.194	0.082	0.467
	20	1.263	0.003	2.786	0.577	0.228	1.718	-0.210	0.981	-0.105	0.563
	70	2.086	-0.174	1.261	0.208	0.449	1.598	-0.226	1.004	0.121	0.476

quantify (in different ways) desirable clustering properties, such as maximal separation between clusters combined with minimal separation within a cluster. The following evaluation methods were used: Davies-Bouldin, Silhouette, Calinski & Harabasz, Baker & Hubert, and Hubert & Levine, from the ClusterSim R package [31]. A higher score indicates a better clustering, except with Davies-Bouldin and Hubert & Levine, where a lower score indicates a better clustering.

Each clustering result was given a score, determined by an ensemble method, where the clustering with the best score for each evaluation measure was given a point, and the clustering with the most points was chosen. Ties were broken by comparing individual scores between the tied instances. For example, between two tied clusterings, the clustering with a better score on an evaluation method such as Davies-Bouldain would be assigned 1 point. This would be repeated for all 5 evaluation methods, and the highest-scoring clustering would be chosen.

4 Results

4.1 Cluster Evaluation

Results from cluster evaluation are shown in Tables 1, 2, and 3. Each line in a table represents a clustering method in combination with the number of variables used to infer the graph (15, 20, or 70). Highlighted numbers represent the best scores for each clustering algorithm and receive 1 point. Total points determine the best clustering. Due to space limitations, results for some graph types are not shown. For self esteem, the high scoring choice was Glasso-Leiden-20, as shown in Table 1. For alcohol risk, as shown in Table 2, both Glasso and MB clusterings performed well, scoring highest on 3 evaluation measures. The tie between Glasso-Leiden-15 and MB-VAT-20 was broken in favor of MB-VAT-20. For HIV results, the Glasso and MB clusterings performed much better than SparCC and kNN, and are shown in Table 3. The 3-way tie was broken in favor

Table 2. Cluster evaluation results for alcohol risk

Cluster Algorithm	#	Glasso					MB				
		Davies-Bouldin	Silhouette	Calinski-Harabasz	Baker-Hubert	Hubert-Levine	Davies-Bouldin	Silhouette	Calinski-Harabasz	Baker-Hubert	Hubert-Levine
Louvain	15	1.624	0.154	10.203	0.700	0.164	1.867	0.011	4.919	0.484	0.272
	20	2.623	0.144	8.020	0.590	0.226	1.813	0.171	11.245	0.643	0.187
	70	3.282	0.024	3.365	0.395	0.338	2.612	0.017	2.151	0.250	0.412
Leiden	15	1.639	0.113	10.475	0.611	0.216	1.893	0.096	-	0.502	0.277
	20	2.175	0.117	8.280	0.518	0.253	2.238	0.120	4.967	0.513	0.265
	70	3.088	-0.030	3.064	0.348	0.369	2.409	0.027	2.282	0.279	0.397
VAT	15	1.659	-0.017	6.069	0.588	0.218	1.807	-0.013	3.898	0.327	0.341
	20	2.271	-0.125	4.348	0.332	0.337	1.364	0.190	10.401	0.627	0.196
	70	2.211	-0.275	1.265	0.095	0.476	2.280	-0.087	1.720	0.186	0.442
Integrity	15	1.659	-0.017	6.069	0.588	0.218	1.622	0.129	3.511	0.648	0.204
	20	1.467	-0.095	3.944	0.544	0.257	1.838	0.042	3.315	0.574	0.241
	70	1.886	-0.251	1.293	0.023	0.514	2.233	-0.104	1.435	0.157	0.467

Table 3. Cluster evaluation results for HIV

Cluster Algorithm	#	Glasso					MB				
		Davies-Bouldin	Silhouette	Calinski-Harabasz	Baker-Hubert	Hubert-Levine	Davies-Bouldin	Silhouette	Calinski-Harabasz	Baker-Hubert	Hubert-Levine
Louvain	15	3.755	-0.065	1.039	0.016	0.547	1.554	0.306	10.891	0.700	0.167
	20	2.387	0.154	10.297	0.578	0.215	2.132	0.164	6.135	0.514	0.270
	70	3.657	0.003	3.146	0.250	0.414	2.637	0.012	2.145	0.202	0.434
Leiden	15	1.600	0.226	20.264	0.747	0.155	1.673	0.292	11.231	0.637	0.189
	20	2.350	0.162	10.774	0.606	0.203	2.150	0.168	6.123	0.481	0.280
	70	3.635	0.016	2.952	0.229	0.431	2.702	0.009	2.151	0.184	0.444
VAT	15	1.422	0.373	37.268	0.783	0.092	1.412	0.098	9.461	0.127	0.353
	20	2.764	-0.214	5.092	0.160	0.405	1.474	0.078	6.408	0.154	0.344
	70	2.952	-0.291	1.332	-0.105	0.576	2.168	-0.156	1.201	-0.100	0.577
Integrity	15	1.364	0.052	12.457	0.732	0.163	1.511	0.246	7.239	0.737	0.155
	20	1.871	-0.142	4.400	0.354	0.328	1.595	0.163	4.405	0.663	0.202
	70	2.713	-0.278	1.304	0.008	0.539	2.168	-0.156	1.201	-0.100	0.577

of Glasso-VAT-15. Based on these results, we visualize and analyze the Glasso-Leiden-20 graph for self esteem, the MB-VAT-20 graph for alcohol risk, and the Glasso-VAT-15 graph for HIV status. The clusterings are described below.

4.2 Cluster Properties

For the top three clusterings, the composition of each cluster for 40 variables, or attributes, is shown in Table 4. The color at the top of the column corresponds to the node color in the network's visualization. Each line of the table represents an attribute, and numbers shown are the percentages of cluster members displaying that attribute. Some variables, like *Income: 0 to 19999* and *In a Relationship* have similar percentages across most of the clusters. On the other hand, the percentage of cluster members who were previously diagnosed with syphilis (*STD: Syphilis*) varies greatly across clusters from 0 to 35%.

Table 4. Cluster composition for 40 variables. Numbers represent the percentage of cluster members exhibiting the described attribute. *Alc.* represents alcohol abuse risk.

Variables/Cluster number	Alc.		HIV+			Self esteem				
	0	1	0	1	2	0	1	2	3	4
Age: 18 to 30	53	100	17	3	5	36	31	28	15	50
Age: 31 to 45	43	0	58	71	82	65	62	46	86	50
Age: 46 to 99	6	0	24	26	14	0	8	28	0	0
Income: 0 to 19999	65	89	84	55	100	58	54	82	72	75
Income: 35000 to 75000	6	11	5	3	0	15	8	10	15	0
Living state: Relatives	8	22	5	3	14	15	24	10	0	0
Living state: Alone	18	22	33	58	5	22	24	10	29	25
Living state: Hotel	3	0	7	0	23	8	8	28	15	0
Living state: Roommates (Friends)	34	0	29	14	36	29	23	9	29	25
Education: Did not finish high school	57	56	50	60	77	37	48	65	87	50
Education: At least some college	36	46	43	34	18	59	47	29	15	50
In a relationship	48	67	40	47	55	29	54	64	43	25
Talks to friends: At least monthly	15	12	13	14	22	16	31	38	15	25
Talks to friends: Yearly	38	11	41	37	27	29	54	37	43	25
Talks to friends: Never	45	78	44	45	41	50	16	28	43	50
Talks to family: At least monthly	29	24	43	46	27	23	16	19	29	25
Talks to family: Yearly	26	23	12	29	14	43	31	37	29	0
Talks to family: Never	29	56	18	11	32	8	31	19	29	75
Feels there is no one to turn to: Most time	16	0	12	20	19	36	39	10	58	0
Feels there is no one to turn to: Sometimes	53	78	53	46	41	36	47	55	15	25
Feels there is no one to turn to: Never	21	22	25	29	23	0	0	10	15	50
Feels left out: Always	5	0	6	15	10	8	8	37	29	25
Feels really understood: At least most time	56	0	49	40	49	87	78	83	58	50
Feels really understood: Never	14	0	21	23	19	0	0	10	0	0
Has trusted friend: At least most time	54	0	41	41	47	87	62	83	86	25
Has trusted friend: Never	19	33	31	26	32	0	0	0	0	25
Has a person to lend money: Always	8	0	16	5	5	29	8	28	0	0
Has a person to lend money: Sometimes	24	56	22	15	23	22	16	10	15	50
Feels Part of a group: At least most time	34	12	39	26	41	58	62	65	58	0
Feels part of a group: Never	34	67	28	40	15	8	0	19	29	50
Feels lack of companionship: Always	11	22	3	6	5	36	16	10	15	0
Feels lack of companionship: Never	18	33	26	26	10	15	16	10	0	75
STD: Syphilis	13	0	39	12	5	0	24	19	15	0
STD: Gonorrhea	15	11	40	3	5	15	31	19	0	25
STD: Genital warts	15	11	36	3	5	22	16	28	0	25
STD: Genital Herpes	8	0	28	3	5	36	0	0	0	25
STD: HIV	17	0	100	100	100	50	31	28	29	25
STD: None	46	67	1	84	100	22	47	46	58	50
Unprotected sex (last 2 months)	13	22	13	3	0	22	24	10	29	50
Unprotected sex (last 12 months)	41	67	30	20	24	43	62	28	29	50
Size of the cluster (number of nodes)	119	9	113	35	22	14	13	11	7	4

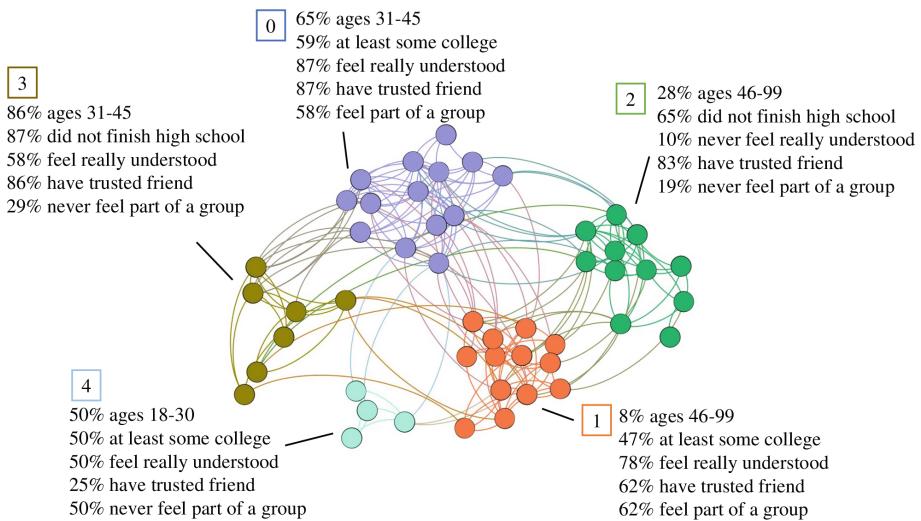


Fig. 1. Probands with high self-esteem are clustered.

Self-esteem. Results for the high self esteem cohort are visualized in Fig. 1. Here probands are divided into 5 clusters, displayed in violet, orange, green, brown, and arctic blue. Cluster 0 is 65% middle-aged, and is well-educated, with 59% of its members having at least some college. This cluster is distinguished by the prevalence of members who feel really understood and have a trusted friend (87%). More than half (58%) of its members feel part of a group of friends at least most of the time. Cluster 1 is somewhat older. 62% of members have a trusted friend and feel part of a group at least most of the time. 47% of its members have at least some college.

Cluster 2 is dominated by 83% of members who have a trusted friend; at the same time, 10% of its members never feel really understood, and 19% never feel part of a group of friends. This cluster is less educated, with 65% not finishing high school. Cluster 3 is distinguished by the middle-aged group, and a large percentage of members who have a trusted friend. More than half (58%) feel really understood, but 29% never feel of part of a group. Cluster 4 is the youngest group, half of whose members have at least some college and feel really understood.

Risk of Alcohol Abuse. The results for risk of alcohol abuse are visualized in Fig. 2 with two clusters in violet and orange. The clustering method is VAT, which produces an attack set, members of which are shown in yellow. Cluster 0 represents the majority of members, most of whom have a low income level. The group is older than cluster 1, with only 53% ages 18–30. 48% are in a relationship. Almost half of the members never talk to friends. Additionally, 17% of members are HIV positive, and 41% have had unprotected sex in the previous year.

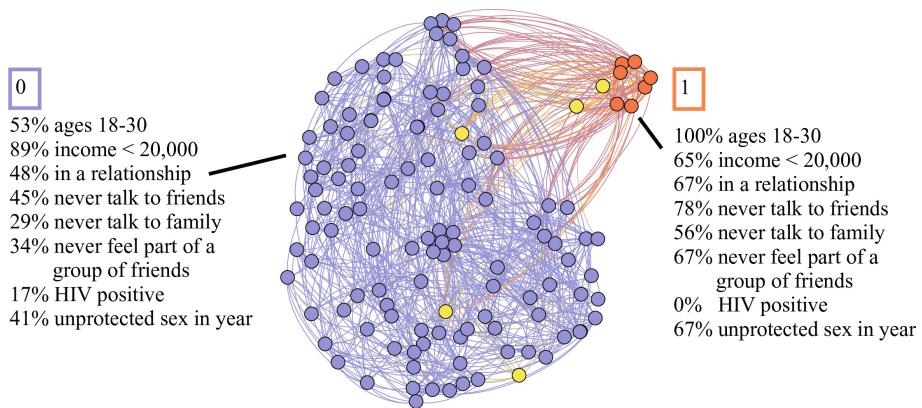


Fig. 2. Probands at risk for alcohol abuse are clustered.

Cluster 1 is a younger group, comprised entirely of 18 to 30 years old, 67% of whom are in a relationship. The cluster is more isolated than cluster 0—78% never talk to friends and more than half never talk to family. 67% had unprotected sex in the last year, but none were HIV positive.

HIV+ Status. The results for HIV positive status are visualized in Fig. 3. The attack set nodes provided by VAT clustering (shown in yellow) along with information from the visualization allowed for enhanced analysis by identifying subgroups for clusters 1 and 2. Cluster 0 is the largest and youngest cluster, with 17% of members aged 18–30. Other clusters consist of members aged 31+. This cluster is primarily distinguished by the prevalence of STDs. For example, 39% have had syphilis and 40% have had gonorrhea. The prevalence of unprotected sex in this group is relatively high at 30% in the last year, and the group is sexually very active, with 13% having unprotected sex within the last 2 months.

Cluster 1 is divided into 4 sub-clusters. Cluster 1b, with many edges into cluster 0, is the only sub-cluster not free of STDs, with only 38% never reporting an STD diagnosis. Cluster 1d, which is attached to the graph by a single edge, truly consists of outsiders. None are in a relationship, and 100% live alone. They have very low rates of unprotected sex. Clusters 1a and 1c show no STDs, but have higher rates of unprotected sex at 38% and 29% within the last year.

Cluster 2 is 100% without STDs, and has relatively low rates of unprotected sex. The subclusters are distinguished primarily by living situation, with cluster 2a living in a hotel, and cluster 2b living with relatives, roommates, or friends.

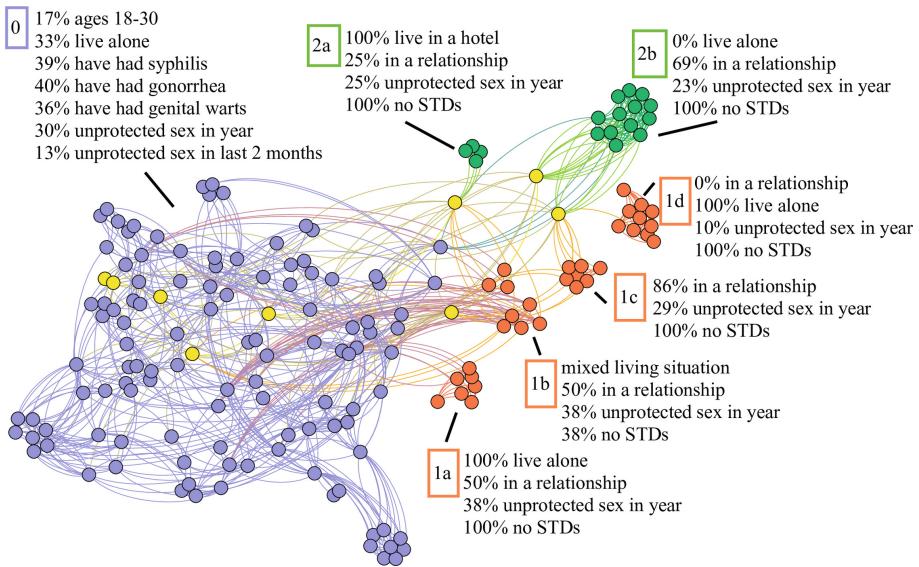


Fig. 3. Probands with HIV positive status are clustered.

5 Discussion and Conclusion

The clustering was able to find meaningful subgroups in the data, and it is particularly interesting that different variable combinations were important for the three targets. It is well-known that feature selection improves machine learning performance, and this study confirms that result. Our best performing data was reduced from hundreds of variables to 15 or 20.

It is hard to pinpoint what factors lead to higher self-esteem; however results show that feeling really understood, having trusted friends, and belonging in a group of friends might lead to higher self-esteem. While these variables were relatively consistent, clusters were distinguished by age and education level.

For risk of alcohol abuse, the important variables were relationships and frequency of communication with family and friends. We identified a younger, more isolated group with a less developed social network and a higher propensity to engage in unprotected sex. From a public health perspective, efforts to prevent HIV targeting this group could also productively attempt to curb alcohol abuse. This is substantiated by many studies linking substance abuse and HIV [9].

For HIV+ status, one cluster had previous history of STDs, and two clusters did not. The connection between HIV and other STDs has been widely studied (e.g. see [5]), and this is another example where public health prevention efforts could be combined. The no-STD clusters were further distinguished by living situation and the frequency of unprotected sex.

From a network science perspective, graphs with the best internal evaluation scores were inferred with Glasso and MB. These methods were developed to

work with microbial count data, which is similar to the attribute counts of the LMSM-CI dataset. There are previous examples of Glasso networks being used with medical data, such as to study PTSD [10] and frailty in older persons [13], but none that we could find with MB, suggesting that this is an area for further research. This result is also surprising in that Glasso and MB performed better than kNN, which is often used to infer graphs in data science applications. For clustering, it is said in [18] that NBR-Clust with VAT is useful for “initial exploratory clustering” and “where the number of desired clusters is low”. Our study confirms that result, as the NBR-Clust with VAT clustering was able to find distinctive, useful clusters. Leiden also performed well. Network science based machine learning and analysis techniques were able to produce results of interest to workers in public health and other health-related areas, and this methodology can be applied to other diverse data sets.

References

1. Abbas, M., et al.: Biomarker discovery in inflammatory bowel diseases using network-based feature selection. *PLoS One* **14**(11), e0225382 (2019)
2. Abirami, M.S., Vennila, B., Chilukalapalli, E.L., Kuriyedath, R.: A classification model to predict onset of smoking and drinking habits based on socio-economic and sociocultural factors. *J. Ambient Intell. Humanized Comput.* **12**(3), 4171–4179 (2020). <https://doi.org/10.1007/s12652-020-01796-4>
3. Ahern, J., Karasek, D., Luedtke, A.R., Bruckner, T.A., van der Laan, M.J.: Racial/ethnic differences in the role of childhood adversities for mental disorders among a nationally representative sample of adolescents. *Epidemiology* **27**(5), 697–704 (2016)
4. Arbia, G., Bramante, R., Facchinetto, S., Zappa, D.: Modeling inter-country spatial financial interactions with graphical lasso: an application to sovereign co-risk evaluation. *Reg. Sci. Urban Econ.* **70**, 72–79 (2018)
5. Barbee, L.A., Khosropour, C.M., Dombrowski, J.C., Golden, M.R.: New HIV diagnosis independently associated with rectal gonorrhea and chlamydia in men who have sex with men. *Sex. Transm. Dis.* **44**(7), 385 (2017)
6. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech. Theor. Exp.* **2008**(10), P10008 (2008)
7. Bolstad, A., Van Veen, B.D., Nowak, R.: Causal network inference via group sparse regularization. *IEEE Trans. Sig. Process.* **59**(6), 2628–2641 (2011). <https://doi.org/10.1109/TSP.2011.2129515>
8. Brugere, I., Gallagher, B., Berger-Wolf, T.Y.: Network structure inference, a survey: motivations, methods, and applications. *ACM Comput. Surv. (CSUR)* **51**(2), 1–39 (2018)
9. Bryant, K.J., Nelson, S., Braithwaite, R.S., Roach, D.: Integrating HIV/AIDS and alcohol research. *Alcohol Res. Health* **33**(3), 167 (2010)
10. Bryant, R.A., et al.: Acute and chronic posttraumatic stress symptoms in the emergence of posttraumatic stress disorder: a network analysis. *JAMA Psychiatry* **74**(2), 135–142 (2017)
11. CDC: Social determinants of health (2021). <https://health.gov/healthypeople/objectives-and-data/social-determinants-health>

12. Clifford, T., Bruce, J., Obafemi-Ajayi, T., Matta, J.: Comparative analysis of feature selection methods to identify biomarkers in a stroke-related dataset. In: 2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), pp. 1–8. IEEE (2019)
13. da Cunha Leme, D.E., da Costa Alves, E.V., Fattori, A.: Relationships between social, physical, and psychological factors in older persons: frailty as an outcome in network analysis. *J. Am. Med. Directors Assoc.* **21**(9), 1309–1315 (2020)
14. Friedman, J., Alm, E.J.: Inferring correlation networks from genomic survey data. *PLoS Comput. Biol.* **8**(9), e1002687 (2012). <https://doi.org/10.1371/journal.pcbi.1002687>
15. Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **9**(3), 432–441 (2008)
16. James, B.T., Luczak, B.B., Grgis, H.Z.: MeShClust: an intelligent tool for clustering DNA sequences. *Nucl. Acids Res.* **46**(14), e83–e83 (2018). <https://doi.org/10.1093/nar/gky315>
17. Kino, S., et al.: A scoping review on the use of machine learning in research on social determinants of health: trends and research prospects. *SSM-Popul. Health* **15**, 100836 (2021). <https://doi.org/10.1016/j.ssmph.2021.100836>. ISSN 2352-8273
18. Kramer, J., Boone, L., Clifford, T., Bruce, J., Matta, J.: Analysis of medical data using community detection on inferred networks. *IEEE J. Biomed. Health Inform.* **24**(11), 3136–3143 (2020)
19. Kurtz, Z.D., Müller, C.L., Miraldi, E.R., Littman, D.R., Blaser, M.J., Bonneau, R.A.: Sparse and compositionally robust inference of microbial ecological networks. *PLoS Comput. Biol.* **11**(5), e1004226 (2015)
20. Matta, J., Ercal, G., Sinha, K.: Comparing the speed and accuracy of approaches to betweenness centrality approximation. *Comput. Soc. Networks* **6**(1), 1–30 (2019). <https://doi.org/10.1186/s40649-019-0062-5>
21. Matta, J., Obafemi-Ajayi, T., Borwey, J., Sinha, K., Wunsch, D., Ercal, G.: Node-based resilience measure clustering with applications to noisy and overlapping communities in complex networks. *Appl. Sci.* **8**(8), 1307 (2018)
22. Matta, J., Zhao, J., Ercal, G., Obafemi-Ajayi, T.: Applications of node-based resilience graph theoretic framework to clustering autism spectrum disorders phenotypes. *Appl. Network Sci.* **3**(1), 1–22 (2018). <https://doi.org/10.1007/s41109-018-0093-0>
23. Meinshausen, N., Bühlmann, P.: High-dimensional graphs and variable selection with the lasso. *Ann. Stat.* **34**(3), 1436–1462 (2006)
24. NIAAA: Drinking levels defined (2017). <https://www.niaaa.nih.gov/alcohol-health/overview-alcohol-consumption/moderate-binge-drinking>
25. Parente, E., Zotta, T., Faust, K., De Filippis, F., Ercolini, D.: Structure of association networks in food bacterial communities. *Food Microbiol.* **73**, 49–60 (2018)
26. Phan, D.V., Yang, N.P., Kuo, C.Y., Chan, C.L.: Deep learning approaches for sleep disorder prediction in an asthma cohort. *J. Asthma* **58**(7), 903–911 (2021)
27. Ramirez-Valles, J.: The protective effects of community involvement for HIV risk behavior: a conceptual framework. *Health Edu. Res.* **17**(4), 389–403 (2002)
28. Seligman, B., Tuljapurkar, S., Rehkopf, D.: Machine learning approaches to the social determinants of health in the health and retirement study. *SSM Popul. Health* **4**, 95–99 (2018)
29. Traag, V., Waltman, L., van Eck, N.J.: From Louvain to Leiden: guaranteeing well-connected communities. arXiv preprint [arXiv:1810.08473](https://arxiv.org/abs/1810.08473) (2018)
30. Udrescu, L., et al.: Clustering drug-drug interaction networks with energy model layouts: community analysis and drug repurposing. *Sci. Rep.* **6**(1), 1–10 (2016)

31. Walesiak, M., Dudek, A.: clustersim package. University of Wraclow, Wraclow (2010). <http://keii.ue.wroc.pl/clusterSim>
32. Yufang, T., Xueming, L., Yan, X., Shuchang, L.: Group lasso based collaborative representation for face recognition. In: 2014 4th IEEE International Conference on Network Infrastructure and Digital Content, pp. 79–83. IEEE (2014)



Network-Based Analysis of Prescription Opioids Dispensing Using Exponential Random Graph Models (ERGMs)

Hilary Aroke¹, Natallia Katenka^{2(✉)}, Stephen Kogut¹,
and Ashley Buchanan¹

¹ Department of Pharmacy Practice, University of Rhode Island, 7 Greenhouse Road, Kingston, RI 02881, USA

{haroke, skogut, buchanan}@uri.edu

² Department of Computer Science and Statistics, University of Rhode Island, 9 Greenhouse Road, Kingston, RI 02881, USA
nkatenka@uri.edu

Abstract. The United States has been experiencing an unprecedented level of opioid overdose-related mortality due in part to excessive use of prescription opioids. Peer-driven network interventions may be beneficial. A key assumption of social network interventions is that of some members of the network act as key players and can influence the behavior of others in the network. We used opioid prescription records to create a social network of patients who use prescription opioid in the state of Rhode Island. The study population was restricted to patients on stable opioid regimens who used one source of payment and received the same opioid medication from ≥ 3 prescribers and pharmacies. An exponential random graph model (ERGM) was employed to examine the relationship between patient attributes and the likelihood of tie formation and modularity was used to assess for homophily (the tendency of individuals to associate with similar people). We used multivariable logistic regression to assess predictors of high betweenness centrality, a measure of influence within the network. 372 patients were included in the analysis; average age was 51 years; 53% were female; 57% were prescribed oxycodone, 34% were prescribed hydrocodone and 9% were prescribed buprenorphine/naloxone. After controlling for the main effects in the ERGM model, homophily was associated with age group, method of payment, number and type of opioid prescriptions filled, mean daily dose, and number of providers seen. Type of opioid and number of prescribers were identified as significant predictors of high betweenness centrality. We conclude that patients who use multiple prescribers or have a diagnosis of opioid use disorder may help promote positive health behaviors or disrupt harmful behaviors in an opioid prescription network.

1 Introduction

The United States is experiencing an unprecedented level of opioid overdose-related deaths. One of the most concerning patterns of utilization involves patients obtaining opioid prescriptions from multiple providers sometimes without the prescribers'

knowledge of other opioid prescriptions being received. [1–3]. Furthermore, some prescribers may engage in prescribing patterns that are no longer recommended for most patient populations [4] such as prescribing a high daily opioid dose, use of combination opioids, and frequent refills. Patients who share the same opioid prescriber usually live in the same neighborhood or community, often in close proximity to each other and their provider. As such, knowledge about an individual prescriber's clinical practices and preferences may be shared indirectly among patients during co-visitation or chance social encounters in the community. A recent study found that healthcare providers tended to share patients with providers who have similar patients in their practice [5]. This suggests that patients utilizing opioids within a single state could be conceptualized as a social network with ties or connections between patients in the network represented by shared provider relationships. We theorized that patients within such a network may exert influence on others' opioid prescription utilization, and patient-based network intervention may impact norms around opioid use and misuse within the network [6].

Limited data suggests that a few high-intensity prescribers play a central role in sustaining the opioid crisis [7, 8]. Understanding the pattern of provider-sharing may help identify influential patients in a network, thereby providing a clearer picture of the paths for dissemination of information within the network. This understanding can inform the implementation of social network interventions designed to improve prescription opioid utilization and reduce the potential for opioid misuse among network members.

A network-based perspective has been used to study a wide range of relational processes involving the flow of information between network members connected to each other. This perspective provides a framework that can be used to understand the structure of a network and how it influences the behavior of individual members in the network [9–11]. Landon et al. recently used network-based methods to demonstrate that characteristics of patient-sharing networks and the position of providers in the network are associated with healthcare resource utilization [6]. Another study has used network analysis to demonstrate racial differences in referral patterns for total hip replacement between communities with low and high concentrations of black residents [12]. These studies support the idea that a central or influential individual holds an important position in their network [13].

The use of social network interventions to promote behavior change has received more attention in recent years but few studies have used social network approaches to address the opioid epidemic. Social network interventions are based on the diffusion of innovations theory and assume that some peers act as role models or champions of change. A recent agent-based model was used to demonstrate the importance of selecting the most effective influential agents using network centrality [14]. There is a dearth of knowledge about characteristics of influential individuals in a network of patients possibly engaged in harmful opioid utilization and methods to identify patients engaged in this kind behavior are limited. This study uses network analysis to evaluate prescription opioid utilization within a single state. The purpose of this study was to identify predictors of influential patients in an opioid prescription network.

2 Methods

2.1 Data Source

We conducted a cross-sectional network-based study using data from the Rhode Island (RI) Prescription Drug Monitoring Program (PDMP) for the 2015 calendar year when the opioid crisis was a major concern in the state. The data contains records of schedule II to IV controlled substances dispensed by all retail pharmacies in the state. It includes de-identified unique patient, prescriber, and dispensing pharmacy information, and a limited number of variables including age (in years) and sex of the patient, National Drug Code (NDC), product name, strength, formulation, and therapeutic class code of the drug plus number of days' supply, metric quantity dispensed, method of payment, and the date each prescription was filled. Daily morphine milligrams equivalents (MME) were estimated using standard conversion factors published by the Center for Disease Control and Prevention.

2.2 Study Population

To minimize the impact of less clinically relevant network connections, we excluded institutional providers and prescribers who issued opioid prescriptions to ≤ 6 patients during the study year. Patients included in the analysis were required to have utilized only one type of opioid medication (i.e., oxycodone, hydrocodone or buprenorphine/naloxone used to treat opioid use disorder); receive opioids from ≥ 3 prescribers and visited ≥ 3 pharmacies and used only one method of payment in order to capture patients who were more likely to be medically stable on opioids and to facilitate meaningful interpretation of the impact of patient attributes.

2.3 Network-Based Framework

PDMP data linked each patient who filled at least one opioid prescription to the prescriber(s). The receipt of one or more opioid prescriptions from a prescriber was used as a proxy for a relationship between a patient and a provider because RI state regulation requires that patients receive a comprehensive evaluation including a physical exam before an opioid medication is prescribed for chronic pain [15]. On the basis of opioid prescription records, a *bipartite* network was constructed and projected into a *simple undirected* patient-based network where all nodes were represented by patients and ties corresponded to shared-provider relationships. Duplicate connections were removed.

2.4 Selecting a Centrality Measure

There are three commonly used centrality measures in social network analysis: degree, closeness and betweenness centrality [13, 16]. Individuals with high degree centrality are seen as important channels of information in a network [17]. These individuals tend to have a large impact on the social norms within the network and are often clustered together in the core of the network, [18, 19] thereby limiting the diffusion of an intervention to peripheral subgroups who might benefit more from the intervention. In an effort to preserve their social status within a network such individuals may be reluctant to change their behavior [20]. Because of these concerns, betweenness and

closeness centrality measures have been advocated for the promotion of health behaviors [16, 21]. As a result, closeness centrality is recommended for use in identifying influential agents when the goal of an intervention is to rapidly spread a *positive* health message to all members of the social network [21]. On the other hand, betweenness centrality is used when the goal is to disrupt the network's ability to spread harmful or *negative* behavior. Providing high betweenness centrality individuals with the intervention knowledge so they no longer connect to other individuals or subgroups reduces network cohesion and slows the spread of negative behaviors in the network. In this study, betweenness centrality was selected as the measure of interest because the goal of a potential opioid prescription network intervention is to prevent the spread opioid misuse, a harmful health behavior.

Network visualizations were selected to optimally place nodes in positions that visually convey important information in the network, [22, 23] and graphical representation was used to show degree distribution. The largest connected component (LCC) was used for the ERGMs and to calculate centrality measures that were standardized for comparison.

2.5 Statistical Network Modeling

ERGM [24] was used to estimate the influence of covariates on the likelihood of tie formation between patients, including homophily (i.e., “birds of a feather flock together”) and differential homophily. Formally a random graph Y consists of a set of nodes and edges; $\{Y_{ij} : i = 1, \dots, n; j = 1, \dots, n\}$, where $Y_{ij} = 1$ if the nodes $\{i, j\}$ are connected and $Y_{ij} = 0$ otherwise. This model can be written as:

$$P(\mathbf{Y} = y | \theta) = \frac{\exp(\theta^T g(y, \mathbf{X}))}{\kappa(\theta, y)}, \quad y \in \mathcal{Y};$$

where θ is a vector of model coefficients and $g(y, \mathbf{X})$ is a vector of statistics based on the adjacency matrix y and covariates \mathbf{X} about the network; and $\kappa(\theta, y)$ is the normalizing factor that ensures a legitimate probability distribution [25, 26].

We evaluated the influence of age or age category, sex, source of payment, and type of opioid medication, number of opioid prescriptions, average daily MME, number of providers, and number of pharmacies. To assess for homophily, we hypothesized that two patients with the same level of a categorical attribute, or similar number of opioid prescriptions filled during the study year, were equally likely to form a network tie.

We limited this analysis to ERGM models that assume dyadic independence of network connections [27]. This assumption specified that patients who had no provider in common were independent but if two patients shared a common provider they are dependent. For model selection, the null and main effects models with and without homophily and differential homophily terms were compared using the log likelihood, the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) [28, 29].

Multivariable logistic regression was used to predict membership in the highest standardized tertile of betweenness centrality. All tests of statistical significance were

two-sided and performed at the 0.05 significance level. Data manipulation was performed with SAS, version 9.4 (SAS Institute, Cary, NC) and network analysis was implemented with R statistical software, version 3.2.3 (R Core Team 2016). The study was approved by the Institutional Review Board at the University of Rhode Island.

3 Results

3.1 Network Description

The full opioid prescription network included in the analysis consisted of 372 patients linked by 746 providers. Table 1 presents a summary of the main demographic and clinical characteristics of all the patients in the full network compared to those in the LCC. The mean age of all patients in the sample was 51 years and 53% were female.

Table 1. Characteristics of Patients in an Opioid Prescription Network of Patients who received prescriptions for oxycodone, hydrocodone or buprenorphine/naloxone from 3 or more prescribers and from 3 or more pharmacies in Rhode Island in 2015 n (%)

Characteristic	Full cohort (N=372)	LCC (N=274)
Age group (years)		
21–44	126 (33.87)	103 (37.59)
45–64	189 (50.81)	134 (48.91)
65+	57 (15.32)	37 (13.50)
Gender		
Female	197 (52.96)	139 (50.73)
Male	175 (47.04)	135 (49.27)
Method of payment for opioids		
Commercial	239 (64.25)	178 (64.96)
Medicare	104 (27.96)	76 (27.74)
Medicaid/cash	29 (7.80)	20 (7.30)
Number of providers		
3	187 (50.27)	134 (48.91)
4+	185 (49.73)	140 (51.09)
Number of pharmacies		
3	257 (69.09)	187 (68.25)
4+	115 (30.91)	87 (31.75)
Type of opioid medication used		
Hydrocodone	126 (33.87)	93 (33.94)
Buprenorphine/naloxone	33 (8.87)	32 (11.68)
Oxycodone	213 (57.26)	149 (54.38)
MME category, mg/day		
<50	197 (52.96)	136 (49.64)
50–90	81 (21.77)	67 (24.45)
90	94 (25.27)	71 (25.91)

Abbreviations: LCC = largest connected component; Rx = prescription(s); MME = morphine milligram equivalent

Of the 5,274 opioid prescriptions that were filled, 57% were for prescriptions of oxycodone, 34% for hydrocodone and 9% for buprenorphine/naloxone. The mean number of opioid prescriptions filled per patient was 14 with 53% receiving <50 MME daily. However, up to 25% of patients had on average >90 MME daily. Most patients paid for all their opioid prescriptions with commercial insurance exclusively. About half of the patients filled opioid prescriptions written by ≥ 4 prescribers and 31% filled their opioid prescriptions at ≥ 4 pharmacies during the one-year study period.

The full projected opioid network contained 372 patients with 1,980 ties among them; 55 connected components; and 32 (8.6%) had no shared providers (isolates). The LCC contained 74% of all the patients in the network including 95.7% of all connections. By visual inspection and examination of network characteristics, the full opioid prescription network and its LCC were very similar (Table 2). The most obvious clustering is seen among patients prescribed buprenorphine-naloxone (Fig. 1). About 85% of patients who shared a provider were connected to other patients who also shared a provider with each of them. Seventy-five of the patients were connected to one or more patients with at least one similar characteristic. The number of shared providers was quite heterogeneous across patients with a tendency for patients with many shared providers to connect to each other, while with fewer shared providers tended to connect with both patients having a lower and a higher number of shared providers consistent with an assortative degree network.

Table 2. Summary of Characteristics of Opioid Prescription Network of Patients Who Received Prescriptions for Oxycodone, Hydrocodone or Buprenorphine/Naloxone from 3 or more Prescribers and from 3 or more Pharmacies in Rhode Island in 2015

Network characteristics	Patient-based network	
Basic characteristics	All	LCC
Number of vertices	372	274
Number of edges	1,980	1,894
Density	0.0287	0.0506
Average degree (SD)	10.7 (11.7)	13.8 (12.0)
Average path length	4.18	4.19
Diameter	10 (13)	10 (13)
Global transitivity	0.85	0.85
Mean Local transitivity	0.73	0.72
Assortative coefficient	0.79	0.75
Number of components	55	1
Size of LCC	274	274

Abbreviations: LCC = largest connected component; SD = standard deviation

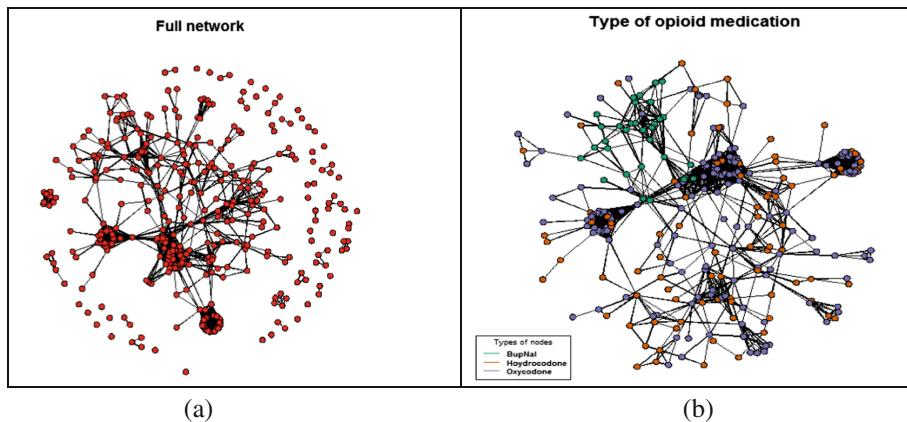


Fig. 1. Two depictions of the largest connected component of a cohort of patients who received prescriptions for oxycodone, hydrocodone or buprenorphine/naloxone from 3 or more prescribers and from 3 or more pharmacies in Rhode Island in 2015: (a) The largest connected component with few clusters (left); (b) clustering by opioid type (right). Abbreviations: BupNal = Buprenorphine/naloxone.

3.2 Predictors of High Betweenness Centrality

Age group, type of opioid received, and number of opioid prescribers were associated with higher betweenness centrality while gender, method of payment, daily MME and number of pharmacies were not. Based on a multivariable logistic regression model only the type of opioid received, and number of opioid prescribers were associated with being classified in the highest tertile of betweenness centrality, after adjusting for other covariates in the model (Table 3). Patients who received opioid prescriptions from ≥ 4 providers were more likely to be classified in the highest tertile than those with only 3 providers. As compared with patients who received hydrocodone, patients prescribed buprenorphine/naloxone were less likely to be classified in the highest degree centrality tertile and more likely to be in the highest betweenness centrality tertile.

Table 3. Odds Ratios (95% CI) Associated with a Patient being Classified in the Upper Tertile of Standardized Betweenness Centrality Measure

Characteristic	Unadjusted OR	Adjusted OR
Number of opioid Rx	0.96 (0.92, 1.01)	0.97 (0.92, 1.03)
Age group (years)		
65+	Ref.	Ref.
21–44	1.90 (0.81, 4.43)	1.65 (0.55, 4.95)
45–64	1.47 (0.64, 3.38)	1.45 (0.52, 4.05)
Gender		
Female	Ref.	Ref.
Male	1.23 (0.74, 2.04)	1.25 (0.73, 2.14)

(continued)

Table 3. (*continued*)

Characteristic	Unadjusted OR	Adjusted OR
Payment method		
Cash	Ref.	Ref.
Commercial	2.19 (0.24, 20.03)	1.86 (0.18, 19.04)
Medicare	1.46 (0.12, 17.23)	2.02 (0.19, 21.43)
Medicaid	1.74 (0.18, 16.39)	1.29 (0.10, 16.93)
Number of providers		
3	Ref.	Ref.
4+	2.02 (1.21, 3.38)	2.60 (1.44, 4.68)
Number of pharmacies		
3	Ref.	Ref.
4+	1.60 (0.93, 2.68)	1.58 (0.89, 2.78)
Opioid type		
Hydrocodone	Ref.	Ref.
Buprenorphine/naloxone	2.92 (1.28, 6.69)	4.33 (1.15, 16.35)
Oxycodone	1.23 (0.69, 2.16)	1.33 (0.65, 2.72)
MME category, mg/day		
<50	Ref.	Ref.
50–90	0.96 (0.51, 1.79)	0.78 (0.36, 1.70)
>90	1.21 (0.66, 2.21)	0.77 (0.28, 2.14)

Abbreviations: CI = confidence interval; OR = odds ratio; Rx = prescription; MME = morphine milligram equivalent.

3.3 Predictors of Network Ties Between Patients

From the ERGM model with main effects only, the total number of opioid prescriptions filled was associated with an increased likelihood of having a shared provider (Table 4). Male patients were less likely to have a shared provider in the network than female patients, although this difference was not statistically significant. Patients who used Medicaid exclusively were less likely to have a shared provider in this opioid prescription network of patients with stable opioid regimens than patients who used commercial insurance alone and the difference was statistically significant. Patients who took either hydrocodone or oxycodone were more likely to have a shared provider in the network. A higher average daily dose of opioids was associated with a greater chance of having a shared provider in the network. Furthermore, patients who had ≥ 4 opioid prescribers in one year were more likely to have at least one shared provider in the opioid prescription network of patients with stable opioid regimens than those with only 3 providers. However, patients who filled their opioid prescriptions at ≥ 4 pharmacies were less likely to have a shared provider in the network than those who used fewer pharmacies.

ERGM model with main effects and homophily terms showed positive and significant parameter estimates for gender, age category, opioid type, average daily MME, and number of providers all indicated the presence of homophily effects for these patient attributes, after controlling for their main effects in the model. All homophily

interaction terms were statistically significant except for method of payment. Use of ≥ 4 pharmacies had no homophilic effects.

Table 4. Main Effects Model to Estimate the Odds Ratios (95% CI) of a Tie Between Two Patients in the Opioid Prescription Network of Patients Using the Largest Connected Component

Term	Odds Ratio (95% CI)
Edges	0.03 (0.02, 0.05) ^a
Number of opioid prescriptions	1.01 (1.00, 1.02)
Gender	
Female	Ref.
Male	1.00 (0.93, 1.07)
Age category, years	
<44	Ref.
45–64	1.07 (0.99, 1.15)
65+	0.54 (0.47, 0.62)
Payment method	
Commercial	Ref.
Medicaid	0.63 (0.53, 0.74)
Medicare	1.14 (1.05, 1.25)
Cash	0.82 (0.64, 1.05)
Opioid type	
Buprenorphine/Naloxone	Ref.
Hydrocodone	2.40 (2.00, 2.87)
Oxycodone	2.67 (2.29, 3.13)
Daily MME, mg/day	
<50	Ref.
50–90	1.70 (1.55, 1.87)
>90	2.02 (1.80, 2.27)
Number of prescribers	
3	Ref.
4+	1.14 (1.06, 1.23)
Number of pharmacies	
3	Ref.
4+	0.96 (0.89, 1.03)

Abbreviations: CI = Confidence interval; MME = morphine milligram equivalent.

^aMultiply reported value by 10^{-1} to get actual odds ratio and 95% confidence interval.

NB: OR > 1 indicate a higher likelihood of sharing a provider with another patient in the network (compared to the reference level for categorical attributes) and OR < 1 indicate lower likelihood.

The ERGM model with main effects and differential homophily terms showed that overall homophilic effects of gender were seen mainly among females; for age mainly among patients 65 years and older; for opioid type mainly among those on buprenorphine/naloxone; and for number of providers mainly among those with few providers in a year, after controlling for other variables in the model, respectively. Adjusted differential homophily effects of payment type, opioid dose and number of pharmacies did not achieve statistical significance. The addition of differential homophily terms did not alter the qualitative effects of the main attributes, except for the subgroup of patients aged 45–64.

4 Discussion

This study suggests that patients in an opioid prescription network are potentially highly connected via shared prescribers. Our sample of 372 patients had 1,980 shared-provider connections and almost 75% of patients were connected to each other either directly or indirectly in one giant component that carried 96% of all connections in the full network. The intensity of prescription opioid utilization is reflected in the large number of prescribers linking patients in the network. Indeed, there were twice as many providers as patients in this prescription opioid network. Although, our analysis was restricted to patients who saw at least 3 providers, more than half the patients saw at least four providers although they appeared to be presumably on a stable opioid regimen. A majority of patients were female and aged 45–64 years old and this is consistent with other published studies [30].

Our opioid prescription network demonstrated evidence of homophily by opioid type, opioid dose, age group, sources of payment and number of providers. On visual inspection, the most obvious clustering was seen among patients prescribed buprenorphine/naloxone. One possible explanation for this clustering is that under the Drug Addiction Treatment Act (DATA) of 2000, only about 70,000 providers³¹ currently have waivers to prescribe buprenorphine/naloxone for patients with OUD. DATA certified providers are more likely to have many patients on prescription opioids in their practice. This may explain why patients receiving buprenorphine/naloxone had fewer shared providers but were 4 times more likely to be classified in the highest betweenness centrality tertile than patients on hydrocodone. This suggests that patients initiated on medication for OUD could be impactful peer educators in their network.

Overall patients with many shared providers were more likely to connect to other patients with a high degree of provider sharing. Prior studies have shown that patients with multiple providers and pharmacies are more likely to engage in patterns of utilization consistent with potential misuse [32–34]. This suggests that attitudes towards opioid misuse could spread within the network. Although older patients with more medical conditions are more likely to receive opioid therapy, they had fewer shared providers compared to younger patients. This may be due to older patients having a more stable health insurance plan and an established primary care provider who meets their medical needs.

Centrality measures suggested that relatively few patients play pivotal roles in this opioid prescription network of patients on stable opioid regimens. Similar conclusions

have been drawn about providers using non-network approaches [7, 8]. If network connections are taken to represent pathways for the flow of information and potential influence, then a measure of how often a patient in the network acts as a bridge between other patients may provide useful information for intervention implementation among patients seeking medical care. Betweenness centrality captures the concept of the influence of each member over the spread of that information designed to alter behavior among others in the network. More information is expected to pass through patients with larger betweenness centrality [19, 35]. This suggests that a public health intervention designed to disrupt harmful behavior in an opioid prescription network of patients with stable opioid regimens who used one source of payment and received the same opioid medication from ≥ 3 prescribers and pharmacies could be provided to patients with high betweenness centrality. This suggests that more information would pass through patients with larger betweenness centralities whose improved pattern of prescription opioid use could disrupt the network cohesion. In practice, patients may not spread information at the same rate and information may not spread through the shortest paths in the network [36]. Despite these limitations, betweenness centrality remains a very useful guide to the potential influence a network member could have over the flow of information and may serve as a useful way to identify patients for prevention and treatment interventions.

For an opioid prescription network, like the one described here, our analysis suggests that such interventions would seek patients with multiple providers or those on medication for opioid use disorder treatment first, particularly those interventions with known spillover effects, such as peer education [37]. These findings support the use of multiple provider episodes as a possible indicator of potential opioid misuse, [38–40] but the use of this definition for patients receiving a medication for OUD has not been previously recognized. Additional studies are needed to evaluate the practical advantages of using betweenness centrality alone as a measure of potential opioid misuse. This can be implemented through sequential analysis of PDMP data in response to the opioid crisis.

Our results suggest that most patients on chronic opioid therapy are connected to each other via provider sharing. Sharing an opioid prescriber may increase the probability of establishing a personal relationship with another patient on opioid therapy through a chance encounter in a physician office or the community because, unlike some other controlled substances, oxycodone, hydrocodone and buprenorphine/naloxone require an office visit for initial prescription and refills. These relationships formed the basis for constructing an opioid prescription network using prescription information captured by the PDMP at the state-level irrespective of the payer. As new state regulations placed quantity limits on opioid prescriptions, patients on chronic opioid therapy will require more frequent office visits for opioid prescription renewals. In an effort to obtain opioid prescriptions for legitimate uses, patients may engage in misuse behaviors because of long waiting times for an appointment, inconvenient office hours, persistence of pain, provider attitude, or absence because of the main provider is on vacation [41–43]. Hence, potential misuse behavior in a patient with persistent chronic pain may be a reflection of fragmented care rather than actual misuse of controlled substances.

Substantial clustering and assortative mixing were driven by patient characteristics. The extent of homophily in this study is similar to what has been observed in other social settings [44, 45]. The cross-sectional nature of our study does not allow us infer the reasons for preferential connections of patients to other patients with similar characteristics. Because certain insurance plans may restrict patients to certain providers in their networks, insurance plan selection may be a partial driver for homophily. One implication of homophily is that it could facilitate the spread of valuable health information through training of influential types of patients in the network and reduce the diffusion of risky behavior around prescription opioid utilization.

The successful application of network science to a systematic problem such as the opioid crisis requires a careful consideration of the choice of nodes and the connections between them to ensure clinical significance. This study demonstrates several advantages of using data from a state PDMP to identify connections within a network of patients receiving prescription opioid therapy. Because the data was collected for use in clinical decision-making at the point of patient care, it provides data that can be used to identify patterns of opioid prescription filling that raise concerns for potential opioid misuse beyond individual patient trajectories.

4.1 Limitations

These analyses are subject to a few limitations. First, provider sharing was used to infer actual relationships between patients when in practice patients may see multiple providers because of fragmented care. The data did not contain information describing geographic proximity, co-visitation to providers, or social interactions in the community. Establishing relationships is difficult in a cross-sectional study like ours. Second, we excluded providers with very few patients and assumed that all prescribers were licensed in the state for the whole calendar year. Their removal may have caused network fragmentation, fundamentally altering some of the properties of the network. Third, our ERGM models assume conditional independence of dyads, which is largely unrealistic in practice [46]. ERGM models of real-world networks tend to perform poorly due time-consuming algorithms, non-converging Markov chains, and associated model degeneracy. The models used in this paper did not include network structures and assume that nodes are fixed and homogenous except for differences captured in nodal attributes. Finally, we note that while pharmacies were required by law to report all controlled substance dispensings to the state's monitoring program, not all pharmacies may have fully complied with this rule and we are unable to know how many opioid dispensings may be missing.

5 Conclusions

Patients utilizing long-term prescription opioids are embedded in a network due to provider-sharing. Characteristics of patients in an opioid prescription network may influence which provider they choose, and patients may have similar norms around opioid use due to provider sharing. Network ties could be leveraged to improve dissemination of health promotion or disruption of harmful behaviors within an opioid

network. Interventions for influential patients in the network may have potential to influence social norms around the use and misuse of prescription opioids that may lead to reductions in prescription opioid-related overdose deaths among patients seeking medical care.

Acknowledgements. We thank the Rhode Island Department of Health for providing the data used for this analysis.

Funding. The project described was supported by Award Number 1DP2DA046856-01 from the National Institute on Drug Abuse of the National Institutes of Health and Drs. Ashley Buchanan and Stephen Kogut were partially supported by Institutional Development Award Number U54GM115677 from the National Institute of General Medical Sciences of the National Institutes of Health, which funds Advance Clinical and Translational Research (Advance-CTR). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Availability of Data and Materials: The data that support the findings of this study are available from Rhode Island Department of Health, but restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available.

References

1. Pradel, V., Thirion, X., Ronfle, E., et al.: Assessment of doctor-shopping for high dosage buprenorphine maintenance treatment in a french region: development of a new method for prescription database. *Pharmacoepidemiol. Drug Saf.* **13**(7), 473–481 (2004)
2. Klienschmidt, R., Price, J., Caught, K.: Doctor shopping. *Aust Fam Physician.* **24**(6), 1037–1041 (1995)
3. Martyres, R.F., Clode, D., Burns, J.M.: Seeking drugs or seeking help? escalating doctor shopping by young heroin users before fatal overdose. *Med. J. Aust.* **180**(5), 211–214 (2004)
4. Dowell, D., Haegerich, T.M., Chou, R.: Cdc guideline for prescribing opioids for chronic pain - united states, 2016. *MMWR Recomm. Rep.* **65**(1), 1–49 (2016)
5. Landon, B.E., Onnella, J.P., Keating, N.L., et al.: Using administrative data to identify naturally occurring networks of physicians. *Med. Care.* **51**(8), 715–721 (2013)
6. Landon, B.E., Keating, N.L., Onnella, J., et al.: Patient-sharing networks of physicians and health care utilization and spending among medicare beneficiaries. *JAMA Intern. Med.* **178**(1), 66–73 (2018)
7. McDonald, D.C., Carlson, K.E.: Estimating the prevalence of opioid diversion by doctor shoppers in the United States. *PLoS ONE* **8**(7), e69241 (2013)
8. Paulozzi, L.J., Strickler, G.K., Kreiner, P.W., et al.: Controlled substance prescribing patterns-prescription behavior surveillance system, eight states, 2013. *MMWR Surveill Summ.* **64**(9), 1–14 (2015)
9. Hall, J.A., Valente, T.W.: Adolescent smoking networks: the effects of influence and selection on future smoking. *Addict. Behav.* **32**(12), 3054–3059 (2007)
10. Ennett, S.T., Bauman, K.E.: Peer group structure and adolescent cigarette smoking: a social network analysis. *J Health Soc Behav.* **34**(3), 226–236 (1993)

11. Rothenberg, R.B., Sterk, C., Toomey, K.E., et al.: Using social network and ethnographic tools to evaluate syphilis transmission. *Sex Transm. Dis.* **25**(3), 154–160 (1998)
12. Ghomrawi, H.M., Funk, R.J., Parks, M.L., Owen-Smith, J., Hollingsworth, J.M.: Physician referral patterns and racial disparities in total hip replacement: a network analysis approach. *PLoS ONE* **13**(2), e0193014 (2018)
13. Valente, T.W.: Network interventions. *Science* **337**(6090), 49–53 (2012)
14. van Woudenberg, T.J., et al.: Identifying Influence Agents That Promote Physical Activity through the Simulation of Social Network Interventions: Agent-Based Modeling Study. *J Med Internet Res.* **21**(8), e12914 (2019)
15. Rules and Regulations for Pain Management, Opioid Use, and the Registration of Distributors of Controlled Substances in Rhode Island [216-RICR-20-20-4]. <https://health.ri.gov/publications/frequentlyaskedquestions/PainMgmtRegs.pdf>. Accessed Aug 2019
16. Borgatti, S.P.: Centrality and network flow. *Soc. Netw.* **27**(1), 55–71 (2005)
17. Freeman, L.C.: Centrality in social networks conceptual clarification. *Soc. Netw.* **1**(3), 215–239 (1978)
18. Medley, A., Kennedy, C., O'Reilly, K., et al.: Effectiveness of peer education interventions for hiv prevention in developing countries: a systematic review and meta-analysis. *AIDS Educ. Prev.* **21**(3), 181–206 (2009)
19. Wasserman, S., Faust, K.: *Social Network Analysis: Methods and Applications*. Vol. 8 Cambridge University Press (1994)
20. Valente, T.W.: *Network Models of the Diffusion of Innovations* (1995)
21. Borgatti, S.P.: Identifying sets of key players in a social network. *Comput. Math. Organ. Theory* **12**(1), 21–34 (2006)
22. Fruchterman, T.M.J. Reingold, E.M.: Graph Drawing by Force-Directed Placement. *Softw. Practice Exp.* **21**(11), 1129–1164 (1991)
23. Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs. *Inf. Process. Lett.* **31**(1), 7–15 (1989)
24. Erdős, P., Rényi, A.: On random graphs. *Publicationes Mathematicae (Debrecen)*. **6**, 290–297 (1959)
25. Frank, O., Strauss, D.: Markov Graphs. *J. Am. Stat. Assoc.* **81**(395), 832–842 (1986)
26. Wasserman, S., Pattison, P.: Logit Models and Logistic Regressions for Social Networks: I. An Introduction to Markov Graphs Andp. *Psychometrika*. **61**(3), 401–425 (1996)
27. Holland, P.W., Leinhardt, S.: An exponential family of probability distributions for directed graphs. *J. Am. Stat. Assoc.* **76**(373), 33–50 (1981)
28. Akaike, H.: Information Theory and an Extension of the Maximum Likelihood Principle. Paper presented at: Second International Symposium on Information Theory (1973)
29. Schwarz, G.: Estimating the dimension of a model. *Ann. Stat.* **6**(2), 461–464 (1978)
30. Aroke, H., Buchanan, A., Wen, X., et al.: Estimating the direct costs of outpatient opioid prescriptions: a retrospective analysis of data from the Rhode island prescription drug monitoring program. *J Manag. Care Spec. Pharm.* **24**(3), 214–224 (2018)
31. Substance Abuse and Mental Health Services Administration (SAMHSA). Medication-Assisted Treatment Program: Practitioner and Program Data. <https://Www.Samhsa.Gov/Medication-Assisted-Treatment/Training-Materials-Resources/Practitioner-Program-Data>. Accessed Aug 2019
32. Substance Abuse and Mental Health Services Administration (SAMHSA). Prescription Drug Monitoring Programs: A Guide for Healthcare Providers. In: Brief, Vol. 10, Issue 1 (2017)
33. Wilsey, B.L., Fishman, S.M., Gilson, A.M., et al.: An analysis of the number of multiple prescribers for opioids utilizing data from the california prescription monitoring program. *Pharmacoepidemiol. Drug Saf.* **20**(12), 1262–1268 (2011)

34. Cepeda, M.S., Fife, D., Chow, W., et al.: Opioid shopping behavior: how often, how soon, which drugs, and what payment method. *J. Clin. Pharmacol.* **53**(1), 112–117 (2013)
35. Keating, N.L., Ayanian, J.Z., Cleary, P.D., et al.: Factors affecting influential discussions among physicians: a social network analysis of a primary care practice. *J. Gen. Intern. Med.* **22**(6), 794–798 (2007)
36. Newman M. Networks: An Introduction: Oxford University Press (2010)
37. Latkin, C.A., Sherman, S., Knowlton, A.: Hiv prevention among drug users: outcome of a network-oriented peer outreach intervention. *Health Psychol.* **22**(4), 332–339 (2003)
38. Barnett, M.L., Olenski, A.R., Jena, A.B.: Opioid-prescribing patterns of emergency physicians and risk of long-term use. *N. Engl. J. Med.* **376**(7), 663–673 (2017)
39. Cepeda, M.S., Fife, D., Berwaerts, J., et al.: Doctor shopping for medications used in the treatment of attention deficit hyperactivity disorder: shoppers often pay in cash and cross state lines. *Am. J. Drug Alcohol Abuse.* **41**(3), 226–229 (2015)
40. Cochran, G., Woo, B., Lo-Ciganic, W.H., et al.: Defining nonmedical use of prescription opioids within health care claims: a systematic review. *Subst. Abus.* **36**(2), 192–202 (2015)
41. Yeung, R.Y., Leung, G.M., McGhee, S.M., et al.: Waiting time and doctor shopping in a mixed medical economy. *Health Econ.* **13**(11), 1137–1144 (2004)
42. Feroni, I., Peretti-Watel, P., Paraponaris, A., et al.: French general practitioners' attitudes and prescription patterns toward buprenorphine maintenance treatment: does doctor shopping reflect buprenorphine misuse? *J. Addict. Dis.* **24**(3), 7–22 (2005)
43. Kasteler, J., Kane, R.L., Olsen, D.M., et al.: Issues underlying prevalence of doctor-shopping behavior. *J. Health Soc. Behav.* **17**(4), 329–339 (1976)
44. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: homophily in social networks. *Ann. Rev. Soc.* **27**(1), 415–444 (2001)
45. Apicella, C.L., Marlowe, F.W., Fowler, J.H., et al.: Social Networks and cooperation in hunter-gatherers. *Nature* **481**(7382), 497–501 (2012)
46. Kolaczyk, E.D., Csárdi, G.: Statistical Analysis of Network Data with R. Springer, New York (2014)



Drug Repositioning Using Multiplex-Heterogeneous Network Embedding: A Case Study on SARS-CoV2

Léo Pio-Lopez^{1,2(✉)}

¹ Aix Marseille Univ, CNRS, Centrale Marseille, I2M, Marseille, France

² Allen Discovery Center, Tufts University, Medford, MA, USA

leo.lopez@tufts.edu

Abstract. Drug repositioning (also called drug repurposing) is a strategy for identifying new therapeutic targets for existing drugs. This approach is of great importance in pharmacology as it is a faster and cheaper way to develop new medical treatments. In this paper, we present, to our knowledge, the first application of multiplex-heterogeneous network embedding to drug repositioning. Network embedding learns the vector representations of nodes, opening the whole machine learning toolbox for a wide variety of applications including link prediction, node labelling or clustering. So far, the application of network embedding for drug repositioning focused on heterogeneous networks. Our approach for drug repositioning is based on multiplex-heterogeneous network embedding. Such method allows the richness and complexity of multiplex and heterogeneous networks to be projected in the same vector space. In other words, multiplex-heterogeneous networks aggregate different multi-omics data in the same network representation. We validate the approach on a task of link prediction and on a case study for SARS-CoV2 drug repositioning. Experimental results show that our approach is highly robust and effective for finding new drug-target associations.

Keywords: Network embedding · Multiplex-heterogeneous network · Multi-layer network · Drug repositioning · Graph representation learning · SARS-CoV2 · COVID-19

1 Introduction

Drug repositioning (also called drug repurposing) is a strategy for identifying new therapeutic targets for existing drugs [34]. This approach is of great importance in pharmacology as it is a faster and cheaper way to develop new medical treatments. It is particularly important for rare diseases - drug discovery is a very expensive process but rare diseases offer a very small market - or in case of a pandemic, when the discovery of a fast treatment is required.

In this paper, we present to the best of our knowledge the first application of multiplex-heterogeneous network embedding for drug repositioning. Network

embedding learns the vector representations of nodes, opening the whole machine learning toolbox for a wide variety of applications including link prediction, node labelling or clustering. In addition, network embedding can handle large-scale networks, composed of millions of nodes [16]. With big data, network embeddings have been applied to a wide range of networks, including for example social, neuronal and molecular networks [25, 30, 31].

So far, the application of network embedding for drug repositioning focused on heterogeneous networks [4, 27, 41]. A heterogeneous network is composed of several monoplex networks with their specific type of nodes and edges and connected by bipartite interactions. In the case of drug repositioning, a heterogeneous network can be composed of drug-drug and target-target networks linked by drug-target bipartite interactions. This kind of biomedical network is usually sparse, noisy and incomplete. It appears fundamental to aggregate different sources of data to reduce the noise and add useful data to the network.

Our approach for drug repositioning is based on multiplex-heterogeneous network embedding. Such a method allows to project in the same vector space the embeddings of both drugs and targets while combining the richness and complexity of multiplex and heterogeneous networks. Indeed, multiplex networks are multi-layer networks, each layer shares the same type of nodes but their edges belong to different types. For drug repositioning, it allows to aggregate different omics data in the same network representation. Recent methods of network embedding for drug repositioning focus on the training of a binary classifier after embedding to do link prediction and find new drug-target or drug-disease associations. In this work, we focus on biological modules by clustering the embeddings. The analysis of the cluster can also lead to the possibility of finding new drug combinations.

We validate the method on a task of link prediction (a standard approach in multi-layer network embedding [1, 32, 43]) and on a case study for SARS-CoV2 drug repositioning. Experimental results show that our approach is highly robust and effective to find new drug-target associations.

2 Related Work

The application of network embedding to drug repositioning is a recent approach. So far, it focused on heterogeneous network embedding [4, 27, 40, 41]. Zhou and colleagues focused on a drug-disease heterogeneous network [44]. The method, called NEDD, applied meta paths of different lengths in the network to explicitly capture the similarities within drugs and diseases, by which they optimize the embeddings of drugs and diseases. NEDD uses a random forest classifier to predict novel associations between drugs and diseases. Other approaches also focused on drug-disease heterogeneous networks like in [41]. In this article, Yang et al. proposed HED to predict potential associations between drugs and diseases based on a drug-disease heterogeneous network using metapath2vec for network embedding [8]. From the embeddings, similarly to [44], they trained an SVM binary classifier to predict new associations. Graph neural networks and graph convolutional networks [42] have

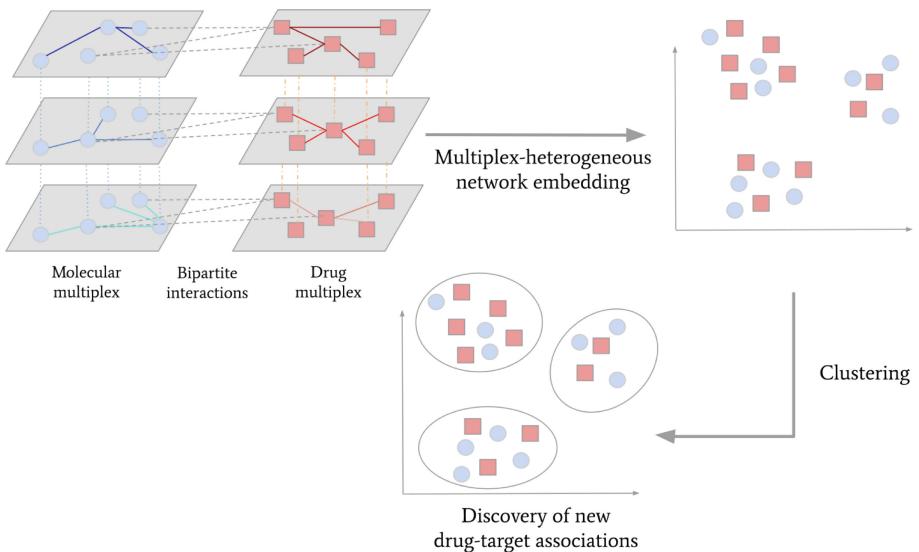


Fig. 1. Upper left: representation of multiplex-heterogeneous network composed of two multiplex networks of three layers. We represented only a simplified versions of bipartite interactions (dashed lines) for a better visual understanding (bipartite interactions of one layer should link each layers of the other multiplex network). The whole figure represents the drug repositioning method based on multiplex-heterogeneous network embedding. Network embedding allows to project drugs (red squares) and targets (blue circles) in a vector space. Then, we apply clustering in order to find new drug-target associations in the clusters.

also been applied to drug repositioning [39] using heterogeneous drug-disease networks. Chen and colleagues introduced a novelty by using cross-network embedding to embed drugs, targets and diseases nodes using two heterogeneous networks, a drug-target and a drug-disease network [4].

To our knowledge, multiplex networks have never been combined with heterogeneous networks for drug repositioning. Multiplex networks are able to integrate multiple types of data and can manage network noise, which impacts the accuracy of the prediction in the case where only heterogeneous networks are used. In addition, our approach differs from the literature by focusing on biological modules to find new drug-target associations. The methods cited above used binary classification to predict new drug-target associations. Our approach focused on a cluster analysis of specific targets to find new drug-target associations (see Fig. 1).

3 Method

We used MultiVERSE for multiplex-heterogeneous network embedding [32]. This method computes the similarities between nodes using random walks with restart on multiplex-heterogeneous network (RWR-MH) and optimizes the embeddings

using Kullback-Leibler minimization. We present these two key components of the method in the next subsections.

3.1 Random Walk with Restart on Multiplex-Heterogeneous Network (RWR-MH)

In a classical random walk (RW), an imaginary particle starts from a seed node, and explores the network, going from nodes to nodes by randomly selecting neighbour with a probability defined by its degree. In the RWR-MH algorithm, it has to travel in the two multiplex networks $G1_M$ and $G2_M$. More formally, the first multiplex network can be defined as $G1_M = (V_M, E_{V_M})$, it is a L_1 -layer multiplex graph, with $n \times L_1$ nodes, $V_M = \{v_i^\alpha, i = 1, \dots, n, \alpha = 1, \dots, L_1\}$. The second multiplex network is a L_2 -layer multiplex graph $G2_M = (U_M, E_{U_M})$ of $m \times L_2$ nodes with $U_M = \{u_j^\alpha, j = 1, \dots, m, \alpha = 1, \dots, L_2\}$. If there is a bipartite edge between two nodes, the particle may jump from a node in one multiplex network to the other multiplex. In multiplex-heterogeneous network, the restart can also happen in different types of node (see Fig. 1) The bipartite graph is defined as $G_B = (V_M \cup U_M, E_B)$ with $E_B \subseteq V_M \times U_M$. With $L_2 > L_1$, we can now define the multiplex-heterogeneous network $G_{MH} = (V_{MH}, E_{MH})$ as:

$$\begin{aligned} V_{MH} &= \{V_M \cup U_M\} \\ E_{MH} &= \left\{ \sum_{\alpha=1, \dots, L_2} E_B^{[\alpha]} \cup E_{V_M} \cup E_{U_M} \right\}. \end{aligned}$$

The different sets of nodes V_M and U_M are only connected by the edges of the bipartite graph. Each layer of the multiplex graphs are connected by bipartite edges.

Therefore, we can now define the evolution of the probability distribution of random walk with restart [26]: $\mathbf{p}_t = (p_t(v))_{v \in V_{MH}}$. This distribution can be described as follows:

$$\mathbf{p}_{t+1}^T = (1 - r)M\mathbf{p}_t^T + r\mathbf{p}_0^T \quad (1)$$

where M denotes a transition matrix that is the column normalization of A_{MH} , which is the multiplex-heterogeneous transition matrix. The vector \mathbf{p}_0 represents the initial probability distribution. With a probability $r \in (0, 1)$, the particle can jump back to the initial node(s) (or seeds), at each step. The probability for the particle to be located at a specific node for an infinite amount of time is represented by the stationary distribution of Eq. (1) [26].

Finally, this distribution can be interpreted as a similarity between the seed(s) and the other nodes. We use it to optimize the embeddings.

3.2 Learning Objective

In the context of MultiVERSE, we need to compute a similarity in the multiplex-heterogeneous network G_{MH} denoted $sim_G : V_{MH} \times V_{MH} \rightarrow \mathbb{R}$ and defined as follows:

$$\forall v \in V_{MH}, \sum_{u \in V} sim_G(v, u) = 1. \quad (2)$$

Therefore, $sim_G(v, \cdot)$, the similarity for any node v of the multiplex-heterogeneous network is expressed as a probability distribution. As $sim_G(v, \cdot)$ is defined as a probability distribution, the normalized similarity distribution in the embedding space is obtained by applying a softmax function. Formally, w_i is defined as the embeddings of node i in the embedding space, the similarity between two nodes embeddings w_u and w_v is defined as the dot product $w_u \cdot w_v^T$ and:

$$sim_{Emb}(v, \cdot) = \frac{\exp(w_v \cdot w^T)}{\sum_{i=1}^n \exp(w_v \cdot w_i)} . \quad (3)$$

The aim of MultiVERSE is to approximate the similarity distribution in the embedding space, denoted $sim_{Emb} : V_{MH} \times V_{MH} \rightarrow \mathbb{R}$ such as $\forall v \in V_{MH}, sim_G(v, \cdot) \approx sim_{Emb}(v, \cdot)$. This learning phase is performed using Kullback-Leibler minimization between the two similarities:

$$\sum_{v \in V_M} KL(sim_G(v, \cdot) \parallel sim_{Emb}(v, \cdot)) \quad (4)$$

By keeping only the terms related to sim_{Emb} as sim_G is constant, we obtain the following objective function:

$$\mathcal{L} = - \sum_{v \in V_M} sim_G(v, \cdot) \log(sim_{Emb}(v, \cdot)) \quad (5)$$

At each iteration, as sim_{Emb} is defined as a softmax function, it is necessary to normalize it over all the nodes of the network, which is computationally heavy. As in the original MultiVERSE algorithm, we used Noise Contrastive Estimation (NCE) to approximate the computations [14].

To sum up, in this framework, the similarity in the multiplex-heterogenous network is computed using Random Walks with Restart on Multiplex-Heterogeneous (RWR-MH) networks [38]. And MultiVERSE applies Kullback-Leibler minimization to optimize the embeddings. The RWR-MH parameters are the same values used in the original study ($r = 0.7, \tau_1 = (1/L_1, 1/L_1, \dots, 1/L_1), \tau_2 = (1/L_2, 1/L_2, \dots, 1/L_2), \delta = 0.5, \lambda = 0.5, \eta = 0.5$) [38].

The reader can refer to the original article for more details on the method and particular implementation of the algorithm [32].

4 Experimental Setup

4.1 Datasets

We used different datasets to construct the multiplex-heterogeneous network. In this work, it is composed of one human molecular multiplex network (3 layers), one drug multiplex network (4 layers) and a heterogeneous drug-target network (corresponding to the bipartite interactions):

- **Human Molecular Multiplex Network with SARS-CoV2 Interactions:** This network is a molecular network, extracted from [32] with additional SARS-CoV2 interactions, composed of 3 layers: 1) A protein-protein interaction (PPI) layer (14704 nodes, 122211 edges) which integrates 4 datasets: Hi-Union, APID (apid.dep.usal.es) (Level 2, human only), Lit-BM (<http://www.interactome-atlas.org/download>) and the additional interactions of SARS-CoV2 from [13] in the PPI layer. 2) The second layer (7926 nodes, 194500 edges) is a pathways layer constructed from the human Reactome data [6] extracted from NDEX [33]. 3) This layer (8537 nodes, 63561 edges) is a molecular complexes layer obtained from the fusion of Hu.map [10] and Corum [12].
- **Drug Multiplex Network:** We used the same dataset as described in [32] with an additional drug layer. The resulting drug multiplex network is composed of the following 4 layers: (1) the first layer (877 nodes, 2795 edges) has been constructed from Bionetdata (<https://rdrr.io/cran/bionetdata/man/DD.chem.data.html>) and the edges correspond to Tanimoto chemical similarities between drugs if superior to 0.6, (2) the second layer (362 nodes, 678 edges) has the edges based on drug combinations as reported in clinical data [5], (3) the third layer (658 nodes, 13397 edges) is the adverse drug-drug interactions network available in [5]. (4) The last layer (4298 nodes, 48593 edges) is the projected drug-target network on the drug layer from Biosnap (<http://snap.stanford.edu/biodata/datasets/10002/10002-ChG-Miner.html>). In this layer, if the Jaccard Index between the neighborhoods of two drug nodes in the original drug-target network is superior to 0.4, an edge between two drugs is created in the projected network. Therefore, two drugs are linked if they share a similar set of genes.
- **Bipartite Network:** We extracted the drug-target bipartite network from [5]. It has 15030 bipartite interactions between 4412 drugs and 2255 protein targets.

4.2 Evaluation of the Approach

Link Prediction. Similarly to [1, 32, 43], we used link prediction to evaluate the quality of the embeddings and validate our approach for drug repositioning. The link prediction pipeline is the following: first, we remove randomly 30% of the bipartite edges to obtain a training network, we then train a Random Forest (as in [32]) on this training network, and test on the 30% removed edges. In order to train the binary classifier, we have to apply operators (Hadamard, Weighted-L1, Weighted-L2, Average and cosine) to the embeddings.

The aim of this validation task is to assess the quality of the embeddings in order to find drug-target associations. Direct comparisons with other methods are not possible as, to our knowledge, there is no other multiplex-heterogeneous network embedding method projecting both types of nodes in the literature.

Case Study on SARS-CoV2 Drug Repositioning. The second approach we used for validation is to test the method on a case study, here the SARS-CoV2 drug repositioning. Indeed, in these times of pandemic, there is abundant literature on the efficacy of different drugs, both *in vitro* and *in vivo*. In addition, with the appearance of variants resistant to vaccines, drug repositioning is still particularly relevant.

In order to find new drug-target associations, we focused on biological modules. Once MultiVERSE has been applied to the drug-target multiplex-heterogeneous network, we used a clustering method on the embeddings and analysed the clusters. This approach allows us to find new drug-target associations but also possible drug combinations as they are included in the same drug-target modules. The clustering method we apply is spherical k-means [3] with $k = 500$ applied on the embedding. We analyse the 27 clusters corresponding to the SARS-CoV2 proteins [13]. We then confront our results to the clinical and biological literature to evaluate the usefulness and quality of our predictions.

5 Experimental Results

5.1 Results on Link Prediction

The ROC-AUC superior to 0.9 with the Hadamard, Average and Cosine operators (see Table 1), meaning that the method can predict with high precision the removed 30% of drug-target links from the corresponding multiplex-heterogeneous networks.

Table 1. ROC-AUC scores for link prediction using MultiVERSE. Link predictions are computed for the bipartite interactions of the multiplex-heterogeneous networks. The scores higher than 0.9 are highlighted in bold.

Operators	ROC-AUC
Hadamard	0.978 ± 0.0006
Weighted-L1	0.708 ± 0.011
Weighted-L2	0.709 ± 0.01
Average	0.962 ± 0.004
Cosine	0.932 ± 0.007

The variance is very small for all operators. The network embedding method is highly robust and steady across each run of the link prediction evaluation test.

5.2 Cluster Analysis of SARS-CoV2 Proteins

In the clusters of the different SARS-CoV2 proteins, we found 88 molecules, out of which 33 are already FDA-approved. Given length constraints, we will present here the most interesting of them.

We found two drugs of interest in the nsp1 protein cluster: cladribine and gallium maltolate. Cladribine is used for multiple sclerosis and has been associated with mild or no symptoms after COVID-19 infection [7]. Gallium maltolate has also in vitro activity against the virus [2].

In the cluster of the nsp6 protein, we have the anti-malarial drug mefloquine that has in vitro inhibition of SARS-CoV2 [11]. Other anti-malarial drugs are present in this cluster like Halofantrine, that have been proposed for COVID-19 repositioning [35] or Voacamidine. Amodiaquine is also a drug that could be a target for this disease [15]. We also found dronedarone that has also been identified as an active inhibitor of the virus [22].

We also found glutathione in the nsp5-C145A protein cluster. This molecule could address the cytokine storm syndrome [18].

In the N protein cluster, we have a repositioning target with S-oxy-L-cysteine. It is a member of the family of L-alpha-amino acids. L-cysteine in combination with vitamin D has been shown to reduce mortality associated with COVID-19 in African Americans [21]. We also found in this cluster a preclinical molecule, the sanglifehrin A, which glue linking IMPDH with cyclophilin A, which itself is involved in viral capsid packaging.

We obtained one interesting drug in the cluster including the orf3b protein of COVID-19. We found the FDA-approved drug ezetimibe as a potential target for repositioning. It has been reported that patients taking this molecule have significantly reduced odds for SARS-CoV-2 hospitalization [20].

In the orf3a protein cluster, we identified the lumichrome molecule, a constituent of honey that would be an inhibitor of the virus [17]. We have also the riboflavin drug which has been shown to inactivate the virus [24].

Several potential targets for repositioning are in the nsp14 protein cluster of COVID-19. We have the AICA ribonucleotide, an AMP-activated protein kinase activator, it has been shown to be effective as a treatment against influenza [29]. In addition, another AMPK activator, metformin, has been associated with decreased mortality in COVID-19 [28]. In this cluster, we also have mycophenolate mofetil which is transformed in its active form the mycophenolic acid as a target for repositioning. This acid is known to inhibit dengue virus and it has been shown that it is active against COVID-19 [23] and has been proposed as a treatment in combination with interferon [9]. We also have the ribavirin anti-viral drug that has been used clinically for COVID-19 treatments in combination with other drugs [19].

The valproic acid is in the cluster of the nsp5 protein of COVID-19. This compound has been proposed as a potential treatment as it reduces ACE2 expression in endothelial cells [36]. We also identified panobinostat, a histone deacetylase inhibitor that could suppress ACE2 and ABO suggesting a preventive effect against COVID-19 [37].

6 Conclusion and Perspectives

We presented the first application of multiplex-heterogeneous network embedding for drug repositioning. The method combines for the first time the richness

of multiplex networks with the complexity of heterogeneous networks in order to find new drug targets. We tested the quality of the embeddings on link prediction and showed the method is highly robust to find drug-target associations for COVID-19. We also found several targets that have anti-viral properties against SARS-CoV2 in vitro or in vivo.

We also identified several other targets for repositioning with this approach. We have for example different anti-cancer drugs like vorinostat or pracinostat in the nsp5 cluster for example that needs further investigations. We also have all the other molecules that are not FDA-approved we included in the multiplex-heterogeneous network.

There are several perspectives to this work. In this article, we projected drugs and targets in the same vector space. An interesting extension would be to project drugs, targets and diseases in the same vector space using either cross-network embedding [4] for multiplex-heterogeneous networks or by extending RWR to 3 multiplex (molecular, drug and disease) and 2 bipartite networks (drug-target and drug-disease for example).

Acknowledgements. We thank Alberto Valdeolivas for discussions on a early phase of this work. We are grateful to Maxime Lucas for helpful comments on a first draft of this work.

References

1. Bagavathi, A., Krishnan, S.: Multi-Net: a scalable multiplex network embedding framework. In: Aiello, L.M., Cherifi, C., Cherifi, H., Lambiotte, R., Lió, P., Rocha, L.M. (eds.) COMPLEX NETWORKS 2018. SCI, vol. 813, pp. 119–131. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05414-4_10
2. Bernstein, L.R., Zhang, L.: Gallium maltololate has in vitro antiviral activity against SARS-COV-2 and is a potential treatment for COVID-19. *Antivir. Chem. Chemother.* **28**, 2040206620983780 (2020)
3. Buchta, C., Kober, M., Feinerer, I., Hornik, K.: Spherical k-means clustering. *J. Stat. Softw.* **50**(10), 1–22 (2012)
4. Chen, H., Cheng, F., Li, J.: idrug: Integration of drug repositioning and drug-target prediction via cross-network embedding. *PLoS Comput. Biol.* **16**(7), e1008040 (2020)
5. Cheng, F., Kovács, I.A., Barabási, A.L.: Network-based prediction of drug combinations. *Nat. Commun.* **10**(1), 1–11 (2019)
6. Croft, D., et al.: The reactome pathway knowledgebase. *Nucleic Acids Res.* **42**(D1), D472–D477 (2014)
7. De Angelis, M., Petracca, M., Lanzillo, R., Morra, V.B., Moccia, M.: Mild or no COVID-19 symptoms in cladribine-treated multiple sclerosis: two cases and implications for clinical practice. *Mult. Scler. Relat. Disord.* **45**, 102452 (2020)
8. Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 135–144 (2017)
9. Dowran, R., et al.: Various interferon (IFN)-inducible transmembrane (IFITM) proteins for COVID-19, is there a role for the combination of mycophenolic acid and interferon? *Biochimie* **177**, 50–52 (2020)

10. Drew, K., et al.: Integration of over 9,000 mass spectrometry experiments builds a global map of human protein complexes. *Mol. Syst. Biol.* **13**(6) (2017)
11. Gendrot, M., et al.: Antimalarial artemisinin-based combination therapies (ACT) and COVID-19 in Africa: in vitro inhibition of SARS-CoV-2 replication by mefloquine-artesunate. *Int. J. Infect. Dis.* **99**, 437–440 (2020)
12. Giurgiu, M., et al.: Corum: the comprehensive resource of mammalian protein complexes-2019. *Nucleic Acids Res.* **47**(D1), D559–D563 (2019)
13. Gordon, D.E., et al.: A SARS-CoV-2 protein interaction map reveals targets for drug repurposing. *Nature* **583**, 459–468 (2020)
14. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: a new estimation principle for unnormalized statistical models. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304 (2010)
15. Hagar, M., Ahmed, H.A., Aljohani, G., Alhaddad, O.A.: Investigation of some antiviral n-heterocycles as COVID 19 drug: molecular docking and DFT calculations. *Int. J. Mol. Sci.* **21**(11), 3922 (2020)
16. Hamilton, W.L., Ying, R., Leskovec, J.: Representation learning on graphs: Methods and applications. *IEEE Data Eng. Bull.* **40**, 52–74 (2017)
17. Hashem, H.E.: In silico approach of some selected honey constituents as SARS-CoV-2 main protease (COVID-19) inhibitors. *Eurasian J. Med. Oncol.* **4**, 196–200 (2020)
18. Horowitz, R.I., Freeman, P.R., Bruzzese, J.: Efficacy of glutathione therapy in relieving dyspnea associated with COVID-19 pneumonia: a report of 2 cases. *Respirat. Med. Case Rep.* **30**, 101063 (2020)
19. Hung, I.F.N., et al.: Triple combination of interferon beta-1b, lopinavir-ritonavir, and ribavirin in the treatment of patients admitted to hospital with COVID-19: an open-label, randomised, phase 2 trial. *Lancet* **395**(10238), 1695–1704 (2020)
20. Israel, A., et al.: Large population study identifies drugs associated with reduced COVID-19 severity. *medRxiv* (2020)
21. Jain, S.K., Parsanathan, R.: Can vitamin D and L-Cysteine Co-supplementation reduce 25(OH)-Vitamin D deficiency and the mortality associated with COVID-19 in African Americans? *J. Am. College Nutr.* **39**(8), 694–699 (2020)
22. Jeon, S., et al.: Identification of antiviral drug candidates against Sars-Cov-2 from FDA-approved drugs. *Antimicrob. Agents Chemother.* **64**(7) (2020)
23. Kato, F., Matsuyama, S., Kawase, M., Hishiki, T., Katoh, H., Takeda, M.: Antiviral activities of mycophenolic acid and IMD-0354 against SARS-CoV-2. *Microbiol. Immunol.* **64**(9), 635–639 (2020)
24. Keil, S.D., Ragan, I., Yonemura, S., Hartson, L., Dart, N.K., Bowen, R.: Inactivation of severe acute respiratory syndrome coronavirus 2 in plasma and platelet products using a riboflavin and ultraviolet light-based photochemical treatment. *Vox sanguinis* **115**(6), 495–501 (2020)
25. Liao, L., He, X., Zhang, H., Chua, T.S.: Attributed social network embedding. *IEEE Trans. Knowl. Data Eng.* **30**(12), 2257–2270 (2018)
26. Lovász, L.: Random walks on graphs: a survey. *Combinatorics Paul Erdos is Eighty 2*(Volume 2), pp. 1–46 (1993). <http://www.cs.yale.edu/publications/techreports/tr1029.pdf>
27. Lu, Z., Wang, Y., Zeng, M., Li, M.: Hnedti: prediction of drug-target interaction based on heterogeneous network embedding. In: *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 211–214. IEEE (2019)
28. Luo, P., et al.: Metformin treatment was associated with decreased mortality in COVID-19 patients with diabetes in a retrospective analysis. *Am. J. Trop. Med. Hygiene* **103**(1), 69–72 (2020)

29. Luzi, L., Radaelli, M.G.: Influenza and obesity: its odd relationship and the lessons for COVID-19 pandemic. *Acta diabetologica* **57**(6), 759–764 (2020)
30. Ma, G., Lu, C.T., He, L., Philip, S.Y., Ragin, A.B.: Multi-view graph embedding with hub detection for brain network analysis. In: 2017 IEEE International Conference on Data Mining (ICDM), pp. 967–972. IEEE (2017)
31. Nelson, W., Zitnik, M., Wang, B., Leskovec, J., Goldenberg, A., Sharan, R.: To embed or not: network embedding as a paradigm in computational biology. *Front. Gen.* **10** (2019)
32. Pio-Lopez, L., Valdeolivas, A., Tichit, L., Remy, É., Baudot, A.: Multiverse: a multiplex and multiplex-heterogeneous network embedding approach. *Sci. Rep.* **11**(1), 1–20 (2021)
33. Pratt, D., et al.: Ndex, the network data exchange. *Cell Syst.* **1**(4), 302–305 (2015)
34. Pushpakom, S., et al.: Drug repurposing: progress, challenges and recommendations. *Nat. Rev. Drug Discov.* **18**(1), 41–58 (2019)
35. Schlagenhauf, P., Grobusch, M.P., Maier, J.D., Gautret, P.: Repurposing anti-malarials and other drugs for COVID-19. *Travel Med. Infect. Dis.* **34**, 101658 (2020)
36. Singh, S., Singh, K.: Valproic acid in prevention and treatment of COVID-19. *Int. J. Respir. Pulm. Med.* **7**(2020)
37. Takahashi, Y., et al.: Histone deacetylase inhibitors suppress ACE2 and abo simultaneously, suggesting a preventive potential against COVID-19. *Sci. Rep.* **11**(1), 1–9 (2021)
38. Valdeolivas, A., et al.: Random walk with restart on multiplex and heterogeneous biological networks. *Bioinformatics* **35**(3), 497–505 (2018)
39. Wang, B., Lyu, X., Qu, J., Sun, H., Pan, Z., Tang, Z.: Gnnd: a graph neural network-based method for drug-disease association prediction. In: 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1253–1255. IEEE (2019)
40. Xiong, Y., et al.: Heterogeneous network embedding enabling accurate disease association predictions. *BMC Med. Genom.* **12**(10), 1–17 (2019)
41. Yang, K., Zhao, X., Waxman, D., Zhao, X.M.: Predicting drug-disease associations with heterogeneous network embedding. *Chaos Interdiscipl. J. Nonlinear Sci.* **29**(12), 123109 (2019)
42. Yu, Z., Huang, F., Zhao, X., Xiao, W., Zhang, W.: Predicting drug-disease associations through layer attention graph convolutional network. *Brief. Bioinform.* **22**(4), bbaa243 (2021)
43. Zhang, H., Qiu, L., Yi, L., Song, Y.: Scalable multiplex network embedding. In: IJCAI, vol. 18, pp. 3082–3088 (2018)
44. Zhou, R., et al.: NEDD: a network embedding based method for predicting drug-disease associations. *BMC Bioinform.* **21**(13), 1–12 (2020)



Drug Repurposing Using Link Prediction on Knowledge Graphs with Applications to Non-volatile Memory

Sarel Cohen¹, Moshik Hershcovitch², Martin Taraz¹, Otto Kißig¹, Andrew Wood^{3(✉)}, Daniel Waddington², Peter Chin³, and Tobias Friedrich¹

¹ Hasso Plattner Institute, Potsdam, Germany

{sarel.cohen,tobias.friedrich}@hpi.de,

{martin.taraz,otto.kissig}@student.hpi.de

² IBM Research, New York, USA

moshikh@il.ibm.com, daniel.waddington@ibm.com

³ Boston University, Boston, USA

{aewood,spchin}@bu.edu

Abstract. The active global SARS-CoV-2 pandemic caused more than 167 million cases and 3.4 million deaths worldwide. The development of completely new drugs for such a novel disease is a challenging, time intensive process and despite researchers around the world working on this task, no effective treatments have been developed yet. This emphasizes the importance of *drug repurposing*, where treatments are found among existing drugs that are meant for different diseases. A common approach to this is based on *knowledge graphs*, that condense relationships between entities like drugs, diseases and genes. Graph neural networks (GNNs) can then be used for the task at hand by predicting links in such knowledge graphs. Expanding on state-of-the-art GNN research, Doshi *et al.* recently developed the DR-COVID model. We further extend their work using additional output interpretation strategies. The best aggregation strategy derives a top-100 ranking of candidate drugs, 32 of which currently being in COVID-19-related clinical trials. Moreover, we present an alternative application for the model, the generation of additional candidates based on a given pre-selection of drug candidates using collaborative filtering. In addition, we improved the implementation of the DR-COVID model by significantly shortening the inference and pre-processing time by exploiting data-parallelism. As drug repurposing is a task that requires high computation and memory resources, we further accelerate the post-processing phase using a new emerging hardware—we propose a new approach to leverage the use of high-capacity Non-Volatile Memory for aggregate drug ranking.

Keywords: Drug repurposing · Knowledge graphs · Link prediction · Collaborative filtering · Non-volatile memory · NVM · MCAS · Python · PyMM

1 Introduction

With the novel coronavirus, a global pandemic with serious socio-economic implications for most parts of our daily lives is active [13]. The limited ability to take precautions for an unsuspected event like this and the rapid spread make finding an effective treatment as necessary as difficult, since the disease-specific knowledge is limited at the beginning and human lives are lost every day. Known and approved drugs happen to be well-studied, thus, they pose a good starting point for swift development of treatments, and an emerging tactic in fighting the pandemic [19]. DrugBank, an extensive database compiling information about drugs approved by the US Food and Drug Administration as well as experimental drugs, contained more than 2 300 approved drugs and over 4 500 experimental drugs as of 2018; both with a strong upward trend [22]. This emphasizes the need for computer aided development of treatments.

Drug repurposing with knowledge graphs, as first described by [1], is the current state-of-the-art approach for finding possible treatments for novel diseases among known drugs using machine learning. Applying drug repurposing allows for a better way to maneuver through the pandemic. It can lead to better treatments for patients infected with one of the COVID-19 strains and a better understanding of the characteristics of the individual strains. Today, we approach the problem of drug repurposing using machine learning, focusing on deep learning methods. The idea of predicting unknown links between entities in a knowledge graph is traditionally known as *Collaborative Filtering*, as described by [17]. In this work we expand on the concept of *graph embeddings*, which map a fixed-size feature vectors to graph nodes and relations. A state-of-the-art technique for the creation of such embeddings based on deep neural networks (DNNs) is TRANSE [2].

Knowledge graph embeddings are already utilized to solve different tasks related to drug discovery, e.g., they are used to predict potential drug targets for diseases to reduce cost and increase speed in the drug development process in general [26]. Regarding the specific application of drug repurposing relying on edge prediction in a knowledge graph of biomedical data (see Sect. 2), [6] present a novel classification approach to this problem by implementing and merging various different ideas and techniques into one ensemble classifier. At its core, they deploy a DNN with an encoder-decoder structure. The encoder mechanism of it, which is based on the *Decagon* graph neural network by [28], was initially proposed for the prediction of side effects of concurrent drug use.

Our Contribution. In this paper we extend the work done by Doshi and Chepuri [4]. Specifically we continue our work in Drug Repurposing [11, 12]. We offer the following contributions to the complex networks community analyzing medicine networks:

1. We improve the post prediction step of [4] by using a clustering of similar diseases and increasing by more than 50% the number of predicted drugs in the top-100 that were or are in clinical trials.

2. We explore the additional application of finding drug candidates similar to a manually pre-selected candidate using collaborative filtering on the same model output. We show that many drugs that are in clinical trial can be found by detecting the drugs that are the most similar (e.g. using cosine-distance on the embedding of the drugs) to a given known drug (or a subset of drugs) which is or was in clinical trials.
3. We re-implement¹ the model described by [4] and improve it by allowing flexible neighborhood capture sizes. We also improve the implementation by [12] by improving training speed, inference time, readability and by reducing pre-processing time from 30 min to 2 min by leveraging matrix operations. We further extend the implementation to support Self-Label-Enhancement.

We also contribute to the *way* drug repurposing is computed. Drug repurposing is a task that requires high computation and memory resources. The emerging hardware of Intel Optane Persistent Memory Modules (Optane-PM) communicates via the memory bus, mitigating bottlenecks such as PCI-express lane availability, using the same interface as DRAM. While there are other Persistent Memory technologies, Optane-PM being the most mature product on the market is based on 3D-XPoint (3DXP) technology and operates at a cache-line granularity with a latency of around 300ns [8], which is more than an order of magnitude faster than the current state of the art NVMe SSDs, but approximately three times slower than DRAM. Additionally, it has high capacity which is 8× larger than the available DRAM—a single DIMM of Optane-PM can reach 512 GB. We note that it is practically necessary to use Optane-PM as the scale of the problem increases [23, 24].

To the best of our knowledge, in this paper, we show for the first time an application of the emerging Optane-PM for the task of Drug Repurposing. We generate a large dataset for the Drug Repurposing problem by extending (both vertically and horizontally) the dataset we have and evaluate two simple aggregation strategies which are implemented and processed on the Optane-PM . We obtain fast and promising results for the use of Optane-PM to process large datasets in the context of Drug Repurposing.

2 Dataset

Our work relies on the Drug Repurposing Knowledge Graph (DRKG) by [7], which compiles data from different biomedical databases. It contains 97,238 entities belonging to 13 entity types and 5,874,261 triplets belonging to 107 edge types. We restrict ourselves to 98 edge types between 4 entity types, namely gene, compound, anatomy and disease, which leaves us with a knowledge graph with 69,036 entities and 4,885,854 edges. In particular, it contains drugs and

¹ Our implementation of the experiments and the model can be found here: <https://drive.google.com/file/d/1hYxMe3AFwcJ4UKsn8SPsZVPW3buXe0u4/view?usp=sharing>.

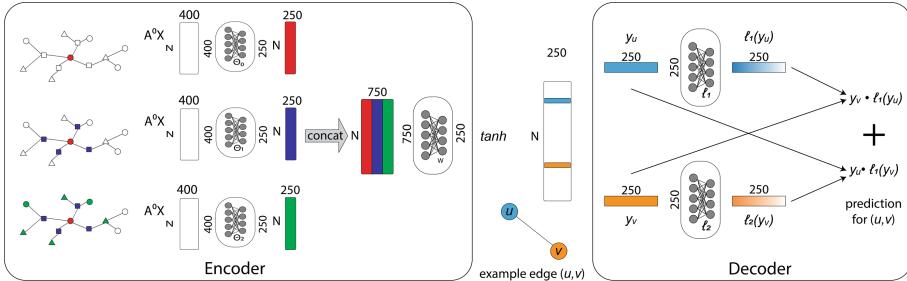


Fig. 1. The architecture of our model as described in Sect. 3.

substances as *compound* entities, as well as different COVID-19 variants as *disease* entities. The edge types include e.g. *compound-treats-disease* edges, which is the kind of edge our model predicts.

One part of DRKG are the precomputed TRANSE embeddings trained using `dgl-ke` by [27]. To train our model to predict whether a given edge in some *compound-treats-disease* relation exists, we have to create suitable training data. To provide our model with both positive and negative samples for training, for each positive edge we sample 30 non-edges in the dataset, which results in a ratio similar to DR-COVID. This process tries to account for the imbalance of edges and non-edges in the ground truth. The set of edges included in the dataset is not complete, however, it is quite certain to be correct. Consequently, the positive edges are given a higher weight in the loss calculation, and the higher number of negative edges (which are not certain to be truly negative) are given a lower weight. To prevent too much imbalance in the individual minibatches, we use a weighted random batch sampler that over-samples the positive samples yielding an expected ratio of 1 : 1.5 of positive to negative samples in each batch.

3 Model Architecture

A Graph Neural Network (GNN) is a message passing framework where vertex embeddings are passed along edges of a graph. A single GNN layer traditionally performs a single round of message passing where messages are transformed via an *edge function*, are collected together into a single message via an *aggregator function*, and finally are used to produce new messages using a *vertex function*. We refer the reader to [10, 18, 28] for a more in-depth description.

In our experiments, we used a traditional encoder-decoder architecture using a two-layer GNN encoder and a custom decoder. The architecture of our model is illustrated in Fig. 1. It consists of a SIGN [5] architecture encoder, which provides an embedding $y \in \mathbb{R}^{250}$ for each node. We apply \tanh to the encoder output and forward it into our decoder. Given two nodes u, v , the decoder takes their encodings y_u, y_v and assigns a score $s_{u,v} \in [0, 1]$, which measures the probability for an edge between nodes u and v to exist. The decoder consists of two linear layers $\ell_1(u)$ and $\ell_2(v)$ that process the encodings y_u and y_v via a sigmoid

function, that is, $\sigma(y_v \cdot \ell_1(y_u) + y_u \cdot \ell_2(y_v))$. The loss of the model is computed using a binary cross entropy loss with logits with weights set as described in Sect. 2.

Implementation. The dataset presents itself as a list of triples, each posing source, relation-type and sink of an edge. This is accompanied by precomputed knowledge graph embeddings. For the preprocessing we first filter out the edges belonging to the part of the knowledge graph we restrict ourselves to. We then construct a graph with the help of DGL [21]. To compute the neighborhood embeddings we feed into the model, we first derive an adjacency matrix $A \in \{0, 1\}^{n \times n}$ from the reduced graph, from which the edges we try to predict, i.e., *compound-treats-disease* edges, have been removed. We then derive the normalized graph Laplacian $\tilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ where $D_{i,i}$ is the degree of node i . Suppose $X \in \mathbb{R}^{n \times 400}$ is the matrix of graph embeddings for the n nodes, then the k th neighborhood is defined as $\tilde{A}^k X$.

4 Output Interpretation

In this section we present different strategies for interpreting the scores that the model outputs for the application of predicting the top- k most promising compound nodes for a given set of disease nodes D . Note that this is important as there are multiple COVID-19 diseases. Let n be the total amount of compound nodes. Predicting all $n \cdot |D|$ edge combinations, our model yields a matrix of scores $S \in \mathbb{R}^{|D| \times n}$. For each of the following strategies we first perform a standardization of the scores per disease using $\hat{s}_{dc} = \frac{s_{dc} - \mu(s_{d*})}{\sigma(s_{d*})}$, where d is the index of a disease in D , c being the index of the compound, $\mu(s_{d*})$ and $\sigma(s_{d*})$ denote the mean and standard deviation over all diseases.

Certain “mild” diseases may be affected by plenty of compounds resulting in those being linked more likely. The standardization helps to achieve a better comparability across different diseases, allowing us to identify the suited compounds for every disease individually and compare those. However, this could also give good scores to some compounds in the case of diseases with no “good” scores in the first place, potentially yielding some less useful proposals.

An aggregation strategy takes our matrix of standardized scores (\hat{s}_{dc}) and derives a list of compounds from it, the top- k of which are our result. We propose the following aggregation strategies. For **global score mean**, we calculate the means of (\hat{s}_{dc}) along axis 0, that is, over all diseases per compound; then we sort the compounds by their respective scores and select the top- k . For **global score maximum**, we find the maxima of (\hat{s}_{dc}) along axis 0; then again we sort the compounds and select the top- k . For **union over disease rankings**, we calculate top- x compounds per disease with x as small as possible such that we get at least k unique compounds in the union. We then concatenate all those top- x lists together to get a top- k compound list.

We also propose **greedy max-min fairness**. Inspired by a game-theoretic approach from auction theory, where we think of the COVID-strains as players

and the compounds as items from which we can only pick a small set, we try to heuristically find a set of compounds that will maximize the COVID strain whose total score is the minimum. Note that Global Score Mean can be considered as allocating the drugs to the COVID strains in a way that obtains the maximum social welfare. In contrast, in the Greedy Max Min Fairness we allocate the candidate drugs among the COVID-strains in a way favoring fairness over social welfare. More precisely, we rank the drugs by iteratively selecting the drug that benefits the disease with the lowest sum of scores over all already selected drugs. From this ordering we then pick the top- k drugs. Because our standardized model outputs \hat{s}_{dc} can be negative, we normalize these by additively shifting them into the positive numbers. This bias however does not interfere with the resulting order because it increases uniformly on all parts of the sum.

Furthermore, in **cluster score maximum**, grouping similar disease types can be used to enhance the accuracy of our top- k predictions. We perform such a grouping using the k-means clustering algorithm. For each cluster, which now represents a group of similar diseases, we use a mean reduction to calculate the score of a compound and then reduce to the maximum across these clusters. A sensible number of clusters to create can be chosen by performing a principal component analysis (PCA) [15] on the standardized scores. Lastly, for **union over cluster rankings**, we perform the top- x selection on clusters calculated with the clustering method described above. This not only allows us to use a greater x because we have fewer lists to pick from, but also to get more consistent top picks because of the internal averages that we apply inside each cluster.

5 Collaborative Filtering

Suppose we already have pre-selected some candidates for clinical trials. Now we would like to identify similar candidates that could be interesting. This new application can be approached using collaborative filtering on our model output. We measure the similarity² along the model's edge predictions per compound.

We test this application by ranking the remaining compounds of our dataset by the cosine similarity to pre-selected candidates. Our pre-selections are sampled randomly from the clinical trial dataset. In the case of one single pre-selected candidate, for selecting the top-100 drugs ranked by similarity to the pre-selected candidate we get a mean of 18 (min. 0, max. 32) hits. Conducting the experiment with 15 pre-selected candidates and selecting drugs corresponding to the top-100 of a global ranking of all similarities yields on average 18 (min. 0, max. 37) hits.

6 Rank Aggregation Using Non-volatile Memory

In this section we demonstrate the use of Non-Volatile Memory for aggregate drug prediction. In general, Optane-PM can perform arbitrary matrix calculations

² To precisely define the cosine similarity between two given drugs i, j , let $\hat{s}_{*i}, \hat{s}_{*j}$ be their prediction scores along the disease dimension. Then their similarity is defined as $\hat{s}_{*i} \cdot \hat{s}_{*j}$.

Table 1. Hits of proposed candidates in actual clinical trials.

Aggregation strategy	# hits
Single disease (median)	20
Global score maximum	22
Global score mean	30
Greedy max min fairness	23
Cluster score maximum with K means ($k = 8$)	18
Cluster score maximum with K means ($k = 3$)	20
Union over disease rankings (Dr-COVID, [4])	21
Union over cluster rankings with K means ($k = 8$)	24
Union over cluster rankings with K means ($k = 3$) [12]	32

while providing significantly more capacity than ordinary DRAM and optionally providing persistence. We chose to use Optane-PM to implement the Global Score Mean and Global Score Maximum aggregation strategies. We chose these strategies for their decent prediction performance (see Table 1) and because they were easy to implement using Optane-PM. We note that we did not select strategies which used clustering due to an incompatibility between scikit-learn [3, 16] (the package clustering was implemented with) and the Optane-PM library. We show that by using Optane-PM, we can process datasets faster than with traditional storage methods such as DRAM + NVMe SSD or memory mapping.

To demonstrate the utility of Optane-PM, we artificially increased the size of the data being operated on. To do so, we extended our ranking matrix of size 2 MB by concatenating entries both vertically and horizontally. Using this scheme, we created data matrices of sizes 33, 66, 131, and 261 GB. This was necessary to show the performance difference between Optane-PM and other storage methods.

6.1 Interacting with Optane-PM

We use a Python 3 library called PyMM to interface with Optane-PM. PyMM has been developed as part of the Memory Centric Active Storage (MCAS) system [20]. PyMM provides a set of abstractions and framework for managing Python variables in locally-attached Optane-PM. For more details regarding MCAS and PyMM, we direct the reader to [20, 24]. Data that is stored in PyMM is persistent and can be accessed and manipulated in-place, directly on device, without requiring a copy or transfer to DRAM. Using PyMM, we store our large data matrices and create aggregate rankings using the two strategies mentioned in Sect. 6.

7 Experiments

7.1 Link Prediction

Our experiments are twofold. We first train our link prediction model to generate probability scores to a candidate edge using an encoder-decoder architecture

described in Sect. 3. We implement the model using PyTorch. We train it using the Adam optimizer [9]. We use 90% of the data for training and the rest for validation. The training is performed on Google Colab utilizing a Nvidia Tesla T4 and it takes ~ 2 min to prepare the graph dataset. We train our model using 25 epochs with a starting learning rate of 10^{-5} and a weight decay of 10^{-2} . Each training epoch took us 30 s, which is a significant improvement over the 610 s of the implementation by [4] and can be attributed to the exploitation of data parallelism we added.

Using this model, we generate prediction matrices by sorting drugs for each covid strain using the learned model. Our first experiment concludes with using aggregation strategies to process the prediction matrix to determine the final drug rankings. These drug rankings are then compared against drugs which are being currently tested in clinical trials.

7.2 Scalable Drug Rank Aggregation

Our second experiment compares Optane-PM against other storage methods. We measure the runtime of using different (simple) aggregation strategies implemented on Optane-PM against their implementations using DRAM. We augment the prediction matrices using the procedure mentioned in Sect. 6 to produce arbitrarily large data. Our experiment compares the following implementations:

- PyMM Implementation.** Prediction matrices are stored on Optane-PM and are processed on device using the Global Score Mean and Global Score Maximum strategies.
- DRAM Implementation.** Prediction matrices are stored entirely on NVMe SSD, and then transferred and processed in DRAM. We note that this implementation is only possible if the machine has sufficient DRAM.
- MMAP_384 Implementation.** Prediction matrices are stored on NVMe SSD. During processing, the required data is loaded from NVMe SSD to DRAM using NumPy’s Memory-Mapping functionality. In this implementation we have 384 GB of DRAM. This configuration allows the entire dataset to be loaded into DRAM, meaning no evictions will occur. Therefore, this is a best-case scenario for memory mapping performance.
- MMAP_64 Implementation.** This implementation is a more realistic memory mapping scenario. While this implementation is almost identical to the previous one, the amount of DRAM has been restricted to 64 GB. This means that the memory mapping routine will need to evict data from DRAM during processing. From a Cloud/infrastructure perspective, this simulates a low-cost machine memory mapping scenario.

Our experiments were conducted on a Lenovo SR650 2U server equipped with two Intel Xeon Gold 6248 (2.5 GHz) processors supporting 80 CPU hardware threads. The server is also equipped with 384 GB (12×32 GB) of DDR4 DRAM and 1.5TB of Optane-PM (12×128 GB) as well as two NVMe SSD disks with 3TB each.

8 Evaluation

8.1 Link Prediction and Drug Aggregation Performance

To test our link prediction model, we compare the top-100 drugs for SARS-CoV2 computed by our learned model to those predicted utilizing the weights of [4]. While their model's top-100 predictions include 22 drugs present in clinical trials, we only reach 15. We suspect the hand-made adjustments to the dataset utilizing undisclosed data sources are responsible for this discrepancy, as this is the sole missing part in our implementation. Consequently, we use their published rankings to measure different aggregation strategies.

To test drug aggregation strategies, we use each strategy to combine rankings of each drug for each covid type to produce a final top-100 ranking. We then compute the number of intersections with the drugs that are currently the subject of clinical trials related to COVID-19 [25]. This information is available on Kaggle as a list of drug names [14].

The results of the different aggregation strategies can be found in Table 1. We see that our Union over Cluster Rankings with KMeans($k=3$) outperforms the other approaches, yielding 32 hits. This is intuitive as using PCA on the prediction scores shows that there are three clusters among the COVID strains.

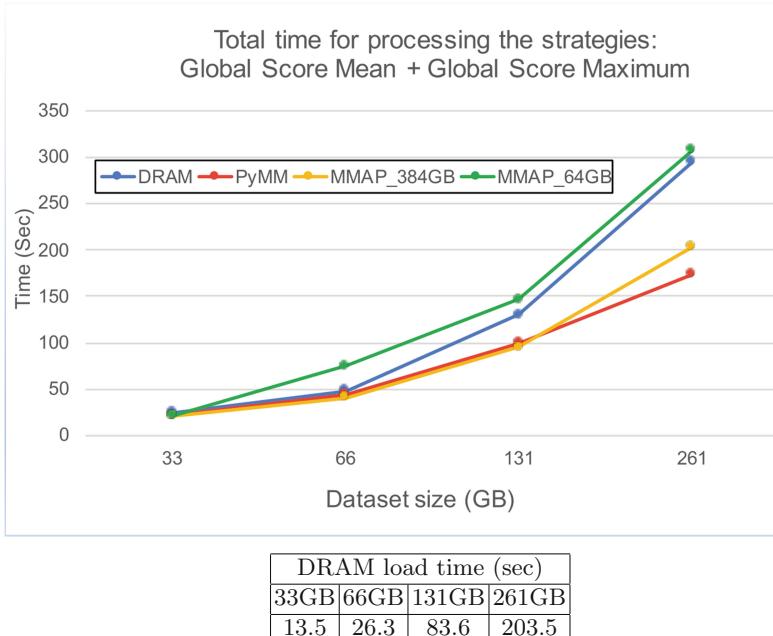


Fig. 2. The total time for processing the two strategies (above): Global Score Mean and Global Score Maximum one after another. The time to copy data from NVMe SSD to DRAM as a function of DRAM size (below).

In contrast, DR-COVID’s aggregation method, Union over Disease Rankings, reaches just 21 hits in our evaluation process.

We observe that hits are not evenly distributed along the rankings of the aggregation strategies, with more hits towards rank 60 and higher, suggesting we are unlikely to get better results by predicting more than the top-100 drugs.

8.2 Scalability Results

Our implementation results can be seen in Fig. 2. In our experiments, Optane-PM always outperformed DRAM. We note that DRAM performance gets significantly worse after consuming 192 GB. This is a result of the dual cpu architecture: DRAM is split between the two sockets in a Non-uniform Memory Access (NUMA) architecture. This means that after 192 GB, data must cross to the other socket which induces a latency penalty of around 100 ns.

We also note that our experiments include the cost of loading data from disk (as needed). The loading time is non-trivial (see the table in Fig. 2). One advantage of Optane-PM is that data is persistent and has no loading time. To measure the cost of compute only, we also tracked the running time after data was loaded. In this case, using Optane-PM is between two and three times slower than DRAM, which is expected since the latency of Optane-PM is known to be approximately three times that of DRAM. We note that the slower latency of Optane-PM is well worth the trade-off for higher capacity as well as persistence.

Surprisingly, the best-case memory mapping implementation performed almost as well as the Optane-PM implementation. This is an artifact of our drug rank aggregation strategies. The two strategies we evaluate are single scan operations, which behaves efficiently using memory mapping. As the memory size increases, we observe the same performance degradation as DRAM due to crossing NUMA node zones.

In a more realistic setting, memory mapping performs the worst. This is due to the eviction policy and DRAM not being able to store the entire dataset. We note that this is also a best-case realistic scenario as once evicted, a row will never be needed again by our strategies. For more advanced strategies, memory mapping will perform significantly worse as multiple passes (sometimes random access) of the data is required.

9 Conclusion

Deep learning can help the development of drugs in the face of a global pandemic. Rather than looking for promising candidates by hand, one can instead rely on graph neural networks. We have been able to clarify the evaluation part of DR-COVID [4] and proposed an aggregation technique yielding better results. Our own implementation improves both training speed as well as readability. We have also shown that using Optane-PM allows researchers to scale techniques efficiently to large datasets, which benefits the drug repurposing community.

References

1. Ashburn, T.T., Thor, K.B.: Drug repositioning: identifying and developing new uses for existing drugs. *Nat. Rev. Drug Discov.* **3**(8), 673–683 (2004)
2. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Neural Information Processing Systems (NeurIPS), pp. 2787–2795 (2013)
3. Buitinck, L., et al.: API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pp. 108–122 (2013)
4. Doshi, S., Prabhakar Chepuri, S.: Dr-COVID: graph neural networks for SARS-CoV-2 drug repurposing. CoRR (2020)
5. Frasca, F., Rossi, E., Eynard, D., Chamberlain, B., Bronstein, M.: Federico Monti. Scalable inception graph neural networks. CoRR, SIGN (2020)
6. Morselli Gysi, D., et al.: Network medicine framework for identifying drug repurposing opportunities for COVID-19. CoRR (2020)
7. Ioannidis, V.N., et al.: DRKG - drug repurposing knowledge graph for COVID-19 (2020). <https://github.com/gnn4dr/DRKG/>
8. Izraelevitz, J., et al.: Basic performance measurements of the intel optane DC persistent memory module. CoRR, abs/1903.05714 (2019). <http://arxiv.org/abs/1903.05714>
9. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016)
11. Kißig, O., Taraz, M., Cohen, S., Doskoč, V., Friedrich, T.: Drug repurposing for multiple COVID strains using collaborative filtering. In: ICLR Workshop on Machine Learning for Preventing and Combating Pandemics (MLPCP@ICLR) (2021)
12. Kißig, O., Taraz, M., Cohen, S., Friedrich, T.: Drug repurposing using link prediction on knowledge graphs. In: ICML Workshop on Computational Biology (CompBio@ICML) (2021)
13. Nicola, M., et al.: The socio-economic implications of the coronavirus pandemic (COVID-19): a review. *Int. J. Surg.* **78**, 185–1939 (2020)
14. Pandey, P.: COVID-19 clinical trials dataset (2021). <https://www.kaggle.com/parulpandey/covid19-clinical-trials-dataset>. Accessed 19 Feb 2021
15. Pearson, K.: F.R.S. LIII. On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Phil. Mag. J. Sci.* **2**, 559–572 (1901)
16. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res* **12**, 2825–2830 (2011)
17. Munir Sarwar, B., Karypis, G., Konstan, J.A., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: International Conference on World Wide Web (2001)
18. Scarselli, F., Gori, M., Chung Tsoi, A., Hagenbuchner, M., Monfardini, G.: The graph neural network model. *IEEE Trans. Neural Netw.* **20**(1), 61–80 (2008)
19. Shah, B., Modi, P., Sagar, S.R.: In silico studies on therapeutic agents for COVID-19: drug repurposing approach. *Life Sci.* **252**(3), 117652 (2020)
20. Waddington, D., Dickey, C., Hershcovitch, M., Seshadri, S.: An architecture for memory centric active storage (MCAS). arXiv preprint [arXiv:2103.00007](https://arxiv.org/abs/2103.00007) (2021)

21. Wang, M., et al.: Deep graph library: a graph-centric, highly-performant package for graph neural networks. arXiv preprint [arXiv:1909.01315](https://arxiv.org/abs/1909.01315) (2019)
22. Wishart, D.S., et al.: Drugbank 5.0: a major update to the drugbank database for 2018. Nucleic Acids Res. 46, D1074–D1082 (2018)
23. Wood, A., Hershcovitch, M., Waddington, D., Cohen, S., Chin, P.: Non-volatile memory accelerated posterior. In: IEEE High Performance Extreme Computing Conference (2021)
24. Wood, A., et al.: Non-volatile memory accelerated geometric multi-scale resolution analysis. In: IEEE High Performance Extreme Computing Conference (2021)
25. World Health Organization: International Clinical Trials Registry Platform (ICTRP) (2021). <https://www.who.int/clinical-trials-registry-platform>. Accessed 19 Feb 2021
26. Ye, C., Swiers, R., Bonner, S., Barrett, I.P.: Predicting potential drug targets using tensor factorisation and knowledge graph embeddings. CoRR, abs/2105.10578 <https://arxiv.org/abs/2105.10578> (2021)
27. Zheng, D., et al.: DGL-KE: training knowledge graph embeddings at scale. In: SIGIR Conference on Research and Development in Information Retrieval (2020)
28. Zitnik, M., Agrawal, M., Leskovec, J.: Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics **34**, i457–i466 (2018)

Urban Systems and Networks



Understanding Imbalance Mechanisms in Shared Mobility Systems

Francisco Prieto-Castrillo^{1,2(✉)}, Rosa M. Benito², and Javier Borondo^{2,3}

¹ Dpto. Electrónica Física, Ing. Eléctrica y Física Aplicada. E.T.S. de Ing. y Sist. de Telecom, Universidad Politécnica de Madrid, Campus Sur, Crtra de Valencia, km 7, 28031 Madrid, Spain

² Grupo de Sistemas Complejos, E.T.S. de Ingeniería Agron., Alim. y de Biosist., Universidad Politécnica de Madrid, Avda. Puerta de Hierro 2-4, 28040 Madrid, Spain

³ Fac. de Ciencias Econ. y Empresariales (ICADE), Dpto. de Gestión Empresarial, Universidad Pontificia Comillas, Calle Alberto Aguilera 23, 28015 Madrid, Spain

Abstract. We explore numerically and analytically how a fleet of vehicles moving through a stations network becomes unbalanced. Framing the system in terms of a mathematical simplex subjected to stochastic flows allows us to understand system's failure rigorously. This allows to find the effect of self-journeys in system's stability. With a birth-death process approach we find analytical upper bounds for random walk and we monitor how the system collapses by super-diffusing under different randomisation conditions.

Keywords: Shared mobility systems · Super-diffusion · Death-birth processes · Complexity

1 Introduction

Actually, it is getting very often to share cars, bicycles or other transportation modes in cities, what it is called Shared Mobility systems (SMs). Vehicles are pulled from and returned to a station network spatially distributed over the city. Whereas public transport displaces large masses of users at regular timings and fixed locations, SMs operate at a smaller scale allowing higher flexibility. The payback of this elasticity is the injection of different amounts of uncertainty into the system. Therefore, understanding the dynamics of SMs and their impact on mobility patterns at the city scale is a matter of primary importance.

On the other hand the problem of mobility in cities has been widely studied [2] and universal laws have been found: despite diversity in travel history, people follow simple reproducible patterns [8, 12]. In particular, bike sharing mobility has been extensively studied, since bikes are sustainable and zero-carbon emission mobility elements [3, 5–7, 9, 11, 13–18].

For example in [3] the authors perform a very descriptive data analysis of the spatio-temporal patterns of bikesharing in Lyon. The identification of patterns allows them to develop an aggregate probability prediction model of system usage (e.g. by forecasting number of rentals). Then they provide data for the

unbalance of the stations network. The authors in [18] find a trimodal activity of morning, rush hour and evening peaks. This pattern persists for several cities (London, Boston, Denver, Washington and Minnesota), which suggests that the trimodal pattern is in fact universal. In their analysis the authors use the trip entropy for the number of A-B trips and find that there is not too much variation in the entropy over a day. Interestingly, in some cases (e.g. Minnesota during weekends) the authors find that self-journeys dominate the whole dynamics. In [15] the benefits of sharing networks are evaluated using data from cabs in NYC. It is shown that cumulative trip lengths can be reduced by 40% or more if trips were shared. Also noticeable in [11] it is shown how no action in the regulation system produces quickly the collapse of the system in terms of depletion/overfull. As the system tends to unbalance, it is necessary the action of an external agent to the system (regulation system) that can operate in times off (static rebalance) or during times on (dynamic rebalance). There is a large amount of related work about how to model and optimize this regulation system. See for instance [6, 11]. Analyzing data from 13 stations in London, the authors in [14] find a bimodal pattern for trips that contrasts with the trimodal one reported in [18]. This is probably due to the fact that in [14] the authors split the user profiles into registered, unregistered, casual, regular and commuter. We conjecture that the aggregated trip frequencies in [14] would render an additional peak between the two maxima in the bimodal pattern. However, more insight into this topic is left for future work.

It is remarkable that in [13] self-organization patterns are observed when bicycles are forced to wander over the network after encountering full stations. However, the authors did not provide any metrics to quantitatively assess the alleged self-organization. In [16] the authors introduce the concept of *shareability* as the fraction of trips that can be shared. How shareable is an A-B route measuring several cities with taxi data. They find a universal scaling law of shareability. Moreover, the authors provide a simple model to account for shareability without the need to adjust parameters. Finally, the birth and death model was used to implement dynamic rebalancing for station occupancy in NYC [4].

In this paper we explore the dynamics of shared mobility systems using simple models following a new approach. Inspired by statistical mechanics we first state the dynamics and collapse conditions as perturbed walks in a simplex. In this sense, we do not go into how to optimize the system, but settle for giving metrics on how the system tends to become unbalanced on its own, as this is necessary to properly understand the dynamics before designing improvement strategies. We validate our approach with data collected from bike sharing. In short, the novelty of the paper is: 1) We give a geometric interpretation of the state of the system and why unbalanced situations occur. This allows us to decouple the dynamics into its fundamental modes. 2) We provide analytical limits for system unbalance, compare with data and find the effect of self-journeys (loops) into the dynamics.

The paper is organized as follows: First in Sect. 2 we introduce the concept of a station simplex that allows us to understand the state of the system in terms of imbalance/energy. Then with this model and a stochastic birth-death process

we find in Sect. 3 analytical bounds for the collapse of the system with a random walk. We also check the situation of a real system with respect to the random walk for different levels of randomness. Also in this section we investigate the effect that loops play on the dynamics of the system. Finally we conclude the work in Sect. 4.

2 Walking in the Simplex: Microstates

For the shared mobility system to be efficient, it must meet two basic objectives: 1) to reach the strategic points of the city and 2) to be efficient in terms of timely available resources. The first requirement is met if the network of stations tries to capture the mobility skeleton of the city. For the second requirement it is necessary to impose additional constraints on the system (timetables, station capacity, etc.). In our model we start from a fleet of N units moving in a set of D stations with capacity C . The number of vehicles at each station is denoted by the quantities $s_i \in \mathbb{N}, i \in D$. Considering the size of the fleet fixed we define the state space Ω as the simplex:

$$\Omega = \{\mathbf{s} \in \mathbb{N}^D \mid \sum_{i=1}^D s_i = N, 0 \leq s_i \leq C\} \quad (1)$$

being \mathbf{s} the D -dimensional occupancy (state) vectors of the system. This simplex is a hyperplane of dimension $D-1$. Given D stations with capacity C the number of possible arrangements for N units can be calculated as the N th coefficient of the following generating function (Eq. 2):

$$G(x) = \sum_{N=0}^{\infty} x^N \left[\sum_{j=0}^D (-1)^j \binom{D}{j} \binom{D+N-1-j(1+C)}{N-j(1+C)} \right] \quad (2)$$

For $D = 3$ and $C = 6$ the series goes like: $G(x) = 3x + 6x^2 + 10x^3 + 15x^4 + 21x^5 + 28x^6 + \dots$ and for $N = 6$ there are 28 possible arrangements (Fig. 1). Since this number grows astronomically with D and N , it is only possible to sample the state space by heuristic methods.

Given the simplex, the trip of one vehicle from station i with occupancy s_i to station j with occupancy s_j is modelled as a directed link $(i, j) \in D \times D$ that occurs when the destination j is incremented by one unit and the origin i is decremented by one unit. The corresponding change of system state in the simplex can be described as:

$$\mathbf{s}' = \mathbf{s} + \Phi(i, j) \quad (3)$$

Where $\phi_k(i, j) \equiv (1 - \delta_{ij})(\delta_{jk} - \delta_{ik})$ are the components of the D -dimensional flow vector $\Phi(i, j)$. The transition is defined as valid (possible) when two conditions hold:

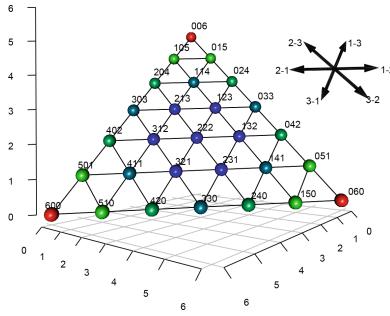


Fig. 1. Walks in the simplex. A toy state space with only $D = 3$ stations, $N = 6$ units with capacity $C = 6$. Each vertex represents a possible state of the network and labels show the occupancies of each station. Colours are proportional to the distance from each point to the barycentre of the simplex located at $(3, 3, 3)$. Arrows in the inset show the possible directions in which states can transit. For example 2–3 means moving a unit from station 2 to station 3. Starting at state $\mathbf{s} = (1, 3, 2)$, the corresponding flow would take us to state $\mathbf{s}' = (1, 2, 3)$.

1. $s_i > 0$ (there is at least one unit at the origin)
2. $s_j \leq C$ (the destination is not full)

Note that a valid transition corresponds to a movement between one state \mathbf{s} and another state \mathbf{s}' in its neighbourhood in the simplex $\mathbb{U}_{\mathbf{s}}$. This neighbourhood is defined as the points that are at a distance equal to $\sqrt{2}$:

$$\mathbb{U}_{\mathbf{s}} = \{\mathbf{s}' \in \Omega \mid \|\mathbf{s} - \mathbf{s}'\| \leq \sqrt{2}\} \quad (4)$$

Starting from the state $\mathbf{s} = (0, 6, 0)$ an example of a valid transition would be $(i, j) = (2, 1)$ (going from station 2 to station 1) which corresponds to $\mathbf{s} = (0, 6, 0) \rightarrow \mathbf{s}' = (1, 5, 0)$. However a transition like $(i, j) = (1, 2)$ is invalid since it would intend to bring a unit from station 1 (which is empty) to station 2 (which is also full). This non-admissible transition would lead to a state outside the simplex (see Fig. 1). Therefore the simplex has the property of being a set of valid states in which the motions are continuous. Moreover, its boundary limit states allow only a certain subset of transitions (i, j) .

In this setting, the farther the system is from the simplex center, the easier it will be to fall into imbalance. This corresponds to the fact that the vehicle system is balanced when station occupancy is close to 50%. This maximizes the likelihood that units will be available and that there will be space to leave a unit. In this way we can quantify the unbalance of the state \mathbf{s} as its distance to the barycenter of the simplex. This leads us to define the energy of a state as: $U(\mathbf{s}) = \|\mathbf{s} - \mathbf{b}\|^2$ where $\mathbf{b} = (C/2, C/2, \dots, C/2)$ is the simplex barycenter. The idea is that points near the barycenter are more balanced and therefore have lower energy. Note that for a valid transition (i, j) we have an energy change:

$$\Delta U = U(\mathbf{s}') - U(\mathbf{s}) = 2(1 - \delta_{ij})(1 + s_j - s_i) \quad (5)$$

So the equi-energetic surfaces are the points of the simplex at equal distance from the barycenter, which correspond to circles in the simplex (inset of Fig. 2). In this case the self-journeys or loops $i = j$ have zero energy change.

2.1 Macrostates

These equi-energetic surfaces in the simplex allow to classify states in classes (macrostates) through the equivalence relation: $\mathbf{s} \sim \mathbf{s}' \iff U(\mathbf{s}) = U(\mathbf{s}')$. The macrostates are thus the equal-balanced surfaces in the quotient set Ω / \sim as defined in [1]. In Fig. 2 we show a schematic of the system dynamics. Vehicle flows between stations (arcs) can be mapped to a walk in the simplex (inset). As these flows are not symmetrical, the system is destabilizing (redder colored zones). This figure also shows the loops that have no energy change in the simplex.

The maximum and minimum energies of the system are calculated from the maximum and minimum distances in the simplex. The lowest energy state is the one closest to the simplex barycentre: $\mathbf{s}_0 = ([C/2], [C/2], \dots, [C/2])$, where $[C/2]$ represents the floor function of $C/2$. Hence the lowest energy of the system is:

$$U^{min} = U(\mathbf{s}_0) = D([C/2] - C/2)^2 \quad (6)$$

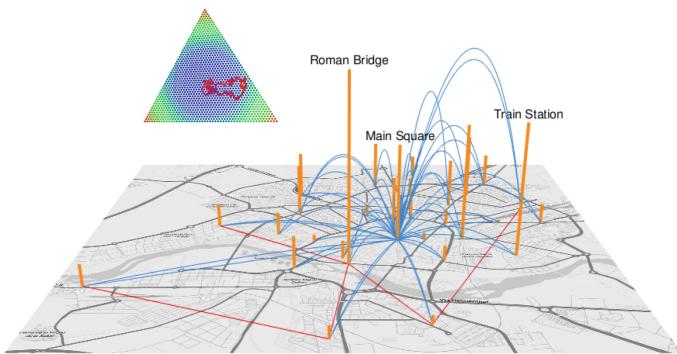


Fig. 2. Schematic of the system dynamics. Vehicle flows between stations (arcs) can be mapped to a walk in the simplex (inset). As these flows are not symmetrical, the system is destabilising (redder coloured zones). This figure also shows the loops (orange bars) that produce no energy change in the simplex since adding and subtracting one unit to the same station produces no net change in the system state \mathbf{s} .

This situation corresponds to balancing the load of all stations in the network with a number of units close to $C/2$.

On the other hand, the unbalance occurs when stations are filled to their limit and others remain empty. If we do the integer division $N = QC + r$ for $Q = 0, \dots, D$ and $r = 0, \dots, C - 1$ a typical unbalanced state will be of the form

given in Eq. 7. This corresponds to Q stations with their maximum occupancy, C , $(D - Q)$ stations completely empty and one station with occupancy r . Thus, it will be an state of the form:

$$\mathbf{s} = (\underbrace{C, C, \dots, C}_{Q \text{ times}}, r, \underbrace{0, 0, 0, \dots}_{D-Q-1 \text{ times}}) \quad (7)$$

and the energy of this state will be:

$$U(\mathbf{s}) = \frac{DC^2}{4} - r(C - r).$$

The maximum¹ is obtained for states in which $r = 0$: $U^{max} = DC^2/4$.

2.2 System Collapse

Let us now consider how likely the system is to collapse as it evolves from an equilibrium situation $\mathbf{s} = \mathbf{s}_0$. The system collapses when at least one station is either completely full (there is no room for an additional deposit) or empty (a user will have to wait to acquire a unit). This can be modelled as a Birth-death chain with two reflecting boundaries as shown in Fig. 3, where the chain goes from 0 to C and its transitions are p (birth), q (death) and l (loop). In this model a generic station i with occupancy $0 \leq s_i \leq C$ decreases by one unit with probability p_{ij} when there is a transition (i, j) with $j \neq i$. On the other hand, s_i grows by one unit with probability p_{ji} when a transition (j, i) has occurred. Thus for each dimension (station) i we have a stochastic process in which the evolution of its occupancy s_i is determined by the transition probabilities:

$$\begin{aligned} \text{birth: } p_i &= \sum_{j \neq i} p_{ji} \\ \text{death: } q_i &= \sum_{i \neq j} p_{ij} \\ \text{loop: } l_i &= 1 - p_i - q_i \end{aligned}$$

The tridiagonal transition matrix of dimension $C + 1 \times C + 1$ in this Birth-death process for s_i is:

$$M = \begin{bmatrix} l_i & p_i & 0 & \dots & 0 \\ q_i & l_i & p_i & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & q_i & l_i \end{bmatrix} \quad (8)$$

The fundamental matrix is calculated from M as: $F = (I - M)^{-1}$. Entries $F_{\mu\nu}$ represent the expected number of times the system is in state ν given that the system started at μ . Therefore, $T(s_i) = \sum_{\nu=1}^{C+1} F_{s_i \nu}$ renders the expected number of times the system is in a transient state (i.e., its lifespan or Mean Absorption Time).

¹ Note that permutations among the s components produce the same energy.

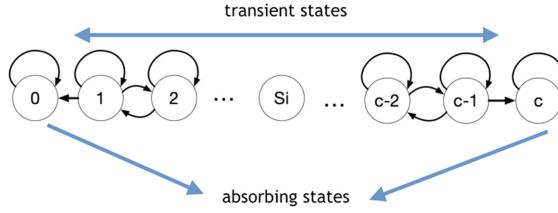


Fig. 3. Birth and Death model for unbalance. Unit flows between stations make the s_i components of \mathbf{s} . In this setting, a generic station i with occupancy $0 \leq s_i \leq C$ decreases by one unit with probability p_{ij} when there is a transition (i, j) with $j \neq i$. On the other hand, s_i grows by one unit with probability p_{ji} when a transition (j, i) has occurred. In this chain the states 0 and C are absorbing states because once one of these states is reached the system collapses.

2.3 Analytical Limit of the Fundamental Matrix for Random Walks

It is interesting to explore the limit of a symmetric probability field: $p_{ij} = 1/D^2$ involving $p_i = q_i = \lambda = (D - 1)/D^2$. In this case F can be inverted analytically in a simple way² [10].

$$F_{ab}^r = \frac{(a + b - |b - a|)(2C - a - b - |b - a|)}{4C\lambda} \quad (9)$$

From here, the random motion absorption times of station i with occupancy s_i are given by: $T^r(s_i) = \sum_{b=1}^{C-1} F_{s_i b}^r$.

To characterize the state of the system we use the minimum lifespan of all stations. By operating we obtain the lifespan limit of the system for random flows:

$$T^r(\mathbf{s}) = \frac{1}{2\lambda} \min_{s_i} \{s_i(C - s_i)\} \quad (10)$$

If we define the energy per station as $u(s_i) \equiv (s_i - C/2)^2$ we can express the expression $T^r(\mathbf{s})$ as:

$$T^r(\mathbf{s}) = \frac{1}{2\lambda} (C^2/4 - \max\{u(s_i)\}) \quad (11)$$

That is, for maximum energies we have minimum absorptions, and conversely, for minimum energies we have maximum absorption times. In other words, when we are in situations of lower energy, the time to system collapse is longer.

3 Assymetry and Super-Difussion. The Effect of Loops

Let's see how the tools defined above help to understand the imbalance in a real case. In particular, we ask how spatio-temporal asymmetries in the distribution

² Also in the non-random case but in a much more complicated way.

of stations and user's schedules create different mobility patterns and how these latter contribute to the overall system imbalance.

Although our analysis is general to any fleet of units, we start from a set of 17k trips stored as time stamped origin-destination records. The data were collected in the period 2014–2017 in the city of Salamanca (Spain). There are 29 stations and as found in [18] we observe in Fig. 4 a trimodal pattern of mobility correlated to the daily working activity. Note that A-B trips have a trimodal pattern on workdays but a bimodal pattern on the weekend. Also, loops have a bimodal pattern for both workdays and weekends. The reason is that loops have a different dynamic than A-B workday trips but not so different than A-B journeys during weekends. This likely caused by different classes of users with different priorities. For example, on weekends people do not need to commute at lunchtime and so the central peak disappears.

On the other hand, the position of the peaks also varies according to work or leisure time schedules. Finally, the cut-off on weekends at 7:00 pm is because the renting time ends at that hour.

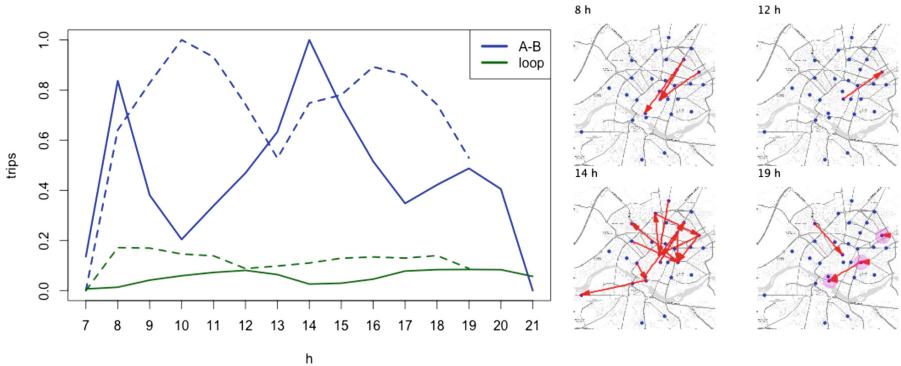


Fig. 4. Mobility patterns for a fleet of 29 bike sharing stations in the city of Salamanca. Left: Temporal evolution of the number of trips (blue) and loops (green) observed during workdays (full line) and weekends (dashed line). Notice the trimodal behavior of the A-B trips during workdays. Right: On-map representation of aggregated fluxes during peak hours. Loops are represented by magenta circles with radius and transparency proportional to their number.

To obtain a global image of the movements at different times we have generated different timestamps (Fig. 4 Right) on the city map indicating movement flows by arrows. To do this, at a given time, we extract the trips and sort them by frequency. Then we generate a uniform distribution between 1 and the maximum number of trips for each hour. Then we choose the paths with a frequency greater than 50% of the paths in that hour. Loops represent only 6.47% of the trips, but as we will see below, they play an important role.

In Fig. 5 we show the Inter-Arrival Times (IAT) that corresponds to a Poisson distribution. The IATs follow a Poisson distribution $P(\tau) = \exp(-\lambda\tau)$ (see

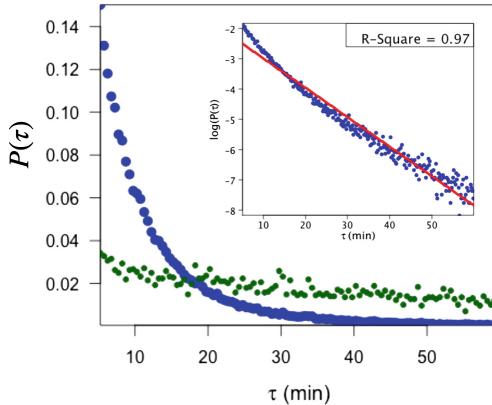


Fig. 5. Inter-ArrivalTimes (IAT) distribution for all trips (blue) and loops (green). Inset: Log-log plot of the distribution for all trips and the corresponding fit to a Poisson distribution (red).

inset in Fig. 5). The arrival rate corresponds to one event every 10 min approx. However when only loops are considered (green dots in Fig. 5) the decay is much slower, almost, linear. This shows that A-B trips and loops create two different time scales. Clearly the spatial distribution of the stations, the temporal patterns of the users and the correlation of these two factors with the number of loops make the dynamics complex. Hence, a question that arises is how to compare in terms of imbalance an observed mobility pattern with respect to a random walk. For this we will use the concept of walk in the simplex presented above along with the results obtained on the mean times to collapse.

The m -length walk in the simplex is constructed by iterating Eq. 3 n steps:

$$\mathbf{s}_n = \mathbf{s}_0 + \sum_{m=1}^n \Phi_m \quad (12)$$

being Φ_m simple random variables taking values $\Phi(i, j)$. By assuming $\mathbf{s}_0 = \mathbf{b}$, energies can be calculated as:³

$$U_n = \sum_{m=1}^n |\Phi_m|^2 + 2 \sum_{a>b}^n \Phi_a \cdot \Phi_b \quad (13)$$

But, noticing that $|\Phi_m|^2 = 2, \forall m$ -this holds from the definition of $\Phi(i, j)$ - the mean value of the energy is:

$$\langle U_n \rangle = 2(n + \sum_{a>b}^n \langle \Phi_a \cdot \Phi_b \rangle) \quad (14)$$

³ If the first state is not exactly the barycentre, there is only one correction as an additive constant.

The super-diffusive term will be 0 for purely random motions. But when there is a non-symmetric flow we will have a higher diffusion than the random one. To see in general how the existing diffusion compares to the real data we start from a state near \mathbf{b} and do a Monte Carlo of walks to absorption or depletion. We then plot the normalized energy in the simplex for each iteration in Fig. 6.

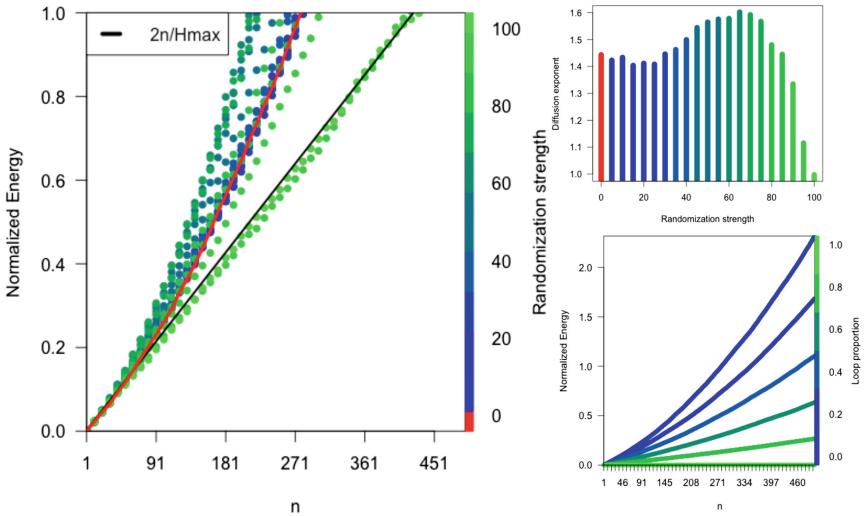


Fig. 6. Super-diffusive behaviour. Left. Normalized energy (Eq. 14 re-scaled between 0 and 1) as a function of the number of steps for different randomisation strengths. The maximum randomization limit is also shown in the black diagonal line. Right-top: diffusion exponent in Eq. 15 for increasing randomization strength. Right-bottom: Effect of loop trips in system diffusion: as the number of loops increases the system becomes less and less superdiffusive.

We have also calculated the diffusion exponents (Fig. 6 right-top) for each randomization level by fitting the curves for each power law of the form:

$$\langle U_n \rangle \sim n^\gamma \quad (15)$$

where n is the number of steps. Note that, as we increase the percentage of random trips, 3 phases appear:

1. <30% random trips: diffusion is less rapid than that found in the data ($\gamma = 1.44$).
2. 30%–90% random trips: the super-diffusive regime spreads faster than that observed in the data and peaks at 65% for $\gamma = 1.60$.
3. >90% random trips (near total random): the super-diffusion drops abruptly and the system starts to diffuse as a random walk in which the normalized energy (unbalance) grows linearly with n and $\gamma = 1$.

On the other hand in Fig. 6 right-bottom we compare the normalized energy as we increase the proportion of loops in the data. As noticed, loops slow down the diffusion and when all trips are loops there is no diffusion at all.

In this way we see that the system naturally overdifuses and that in general external mechanisms are necessary to rebalance the system as is common in shared mobility. As stressed, this super diffusion is due to the asymmetric drift of the p_{ij} . The macroscopic effect of this mechanism must also have an origin in long correlations due to interactions between the parts of the system.

4 Conclusion

In this paper we have analyzed both analytically and numerically the dynamics of a set of vehicles in a finite capacity network of stations. According to previous results we have found that the system tends to collapse naturally. In particular, the diffusion is anomalous because the asymmetry of travel trends creates diffusions that are not balanced in different areas of the city. Therefore an additional external rebalancing mechanism is necessary to keep the system in equilibrium. Furthermore, we have given analytical expressions for the collapse probability and for the super-diffusion exponents. On the other hand, we have found that the loop trips, although being a small proportion of the total trips, play an important role in the stabilization of the system. These results have been compared with real data from a bike-sharing fleet in the city of Salamanca (Spain). Finally, what we have done, even applied to bicycles, gives us universal models of general applicability.

Acknowledgment. The authors acknowledge support from Project No. PGC2018-093854-B-I00 of the Spanish Ministerio de Ciencia Innovación y Universidades of Spain.

References

1. Atmanspacher, H.: On macrostates in complex multi-scale systems. *Entropy* **18**(12), 426 (2016)
2. Barbosa, H., et al.: Human mobility: models and applications. *Phys. Rep.* **734**, 1–74 (2018). <https://doi.org/10.1016/j.physrep.2018.01.001>
3. Borgnat, P., Abry, P., Flandrin, P., Robardet, C., Rouquier, J.B., Fleury, E.: Shared bicycles in a city: a signal processing and data analysis perspective. *Adv. Complex Syst.* **14**(03), 415–438 (2011). <https://doi.org/10.1142/S0219525911002950>
4. Chiariotti, F., Pielli, C., Zanella, A., Zorzi, M.: A dynamic approach to rebalancing bike-sharing systems. *Sensors (Switzerland)* **18**(2), 1–22 (2018)
5. Çolak, S., Lima, A., González, M.C.: Understanding congested travel in urban areas. *Nat. Commun.* **7**, 10793 (2016)
6. Crisostomi, E., Faizrahnemoon, M., Schlote, A., Shorten, R.: A Markov-chain based model for a bike-sharing system. In: Proceedings 2015 International Conference on Connected Vehicles and Expo, ICCVE 2015, pp. 367–372 (2015)
7. Fricker, C., et al.: Mean field analysis for inhomogeneous bike sharing systems. to cite this version : Mean Feld Analysis for Inhomogeneous Bike Sharing Systems, AofA, July 2012, Montreal, Canada, pp. 0–12 (2014)

8. Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.L.: Understanding individual human mobility patterns. *Nature* **453**(7196), 779–782 (2008)
9. Hamon, R., Borgnat, P., Flandrin, P., Robardet, C.: Networks as signals, with an application to a bike sharing system. In: 2013 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2013 - Proceedings, pp. 611–614 (2013)
10. Hu, G.Y., O'Connell, R.F.: Analytical inversion of symmetric tridiagonal matrices. *J. Phys. A Math. Gen.* **29**(7), 1511–1513 (1996)
11. Labadi, K., Benarbia, T., Barbot, J.P., Hamaci, S., Omari, A.: Stochastic petri net modeling, simulation and analysis of public bicycle sharing systems. *Auto. Sci. Eng. IEEE Trans.* **12**(99), 1–16 (2014). <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6869036>
12. Noulas, A., Scellato, S., Lambiotte, R., Pontil, M., Mascolo, C.: A tale of many cities: universal patterns in human urban mobility. *PLoS ONE* **7**(5), e37027 (2012). <https://doi.org/10.1371/journal.pone.0037027>
13. Preisler, T., Dethlefs, T., Renz, W.: Self-organizing redistribution of bicycles in a bike-sharing system based on decentralized control. In: 2016 Federated Conference on Computer Science and Information Systems (FedCSIS), vol. 8, pp. 1471–1480 (2016). <https://fedcsis.org/proceedings/2016/drps/126.html>
14. Purnama, I.B.I., Bergmann, N., Jurdak, R., Zhao, K.: Characterising and predicting urban mobility dynamics by mining bike sharing system data. In: Proceedings - 2015 IEEE 12th International Conference on Ubiquitous Intelligence and Computing, 2015 IEEE 12th International Conference on Advanced and Trusted Computing, 2015 IEEE 15th International Conference on Scalable Computing and Communications, vol. 20, pp. 159–167 (2015)
15. Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S., Ratti, C.: Quantifying the benefits of vehicle pooling with shareability networks. *Proc. Natl. Acad. Sci. USA* **111**(37), 13290–13294 (2013). <http://arxiv.org/abs/1310.2963%0Adx.doi.org/10.1073/pnas.1403657111>
16. Tachet, R., Sagarra, O., Santi, P., Resta, G., Szell, M., Strogatz, S.H., Ratti, C.: Scaling law of urban ride sharing. *Sci. Rep.* **7**, 1–6 (2017). <http://dx.doi.org/10.1038/srep42868>
17. Vazifeh, M.M.: Addressing the minimum fleet problem in on-demand urban mobility. *Nature* **557**, 534–538 (2018)
18. Zaltz Austwick, M., O'Brien, O., Strano, E., Viana, M.: The structure of spatial networks and communities in bicycle sharing systems. *PLoS ONE* **8**(9) (2013). <http://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0074685&type=printable>



Neighborhood Discovery via Network Community Structure

Aaron Bramson^{1,2,3(✉)}

¹ GA Technologies, Inc., Roppongi 3-2-1, Minato-ku, Tokyo 106-6290, Japan
a_bramson@ga-tech.co.jp

² RIKEN Center for Biosystems Dynamics Research, Laboratory for Symbolic Cognitive Development, Minatojima-Minamimachi 6-7-3, Chuo-ku, Kobe 650-0047, Japan

³ Department of General Economics, Ghent University, Tweekerkenstraat 2, 9000 Ghent, Belgium

Abstract. Compared to social and information networks, the geospatial characteristics of transportation networks make them structurally constrained. Although road, flight, train, and other such networks have been analyzed using social network analysis methods, the results typically fail to capture useful characteristics or make informative comparisons. In the case of road networks, natural constraints on the edge distribution weaken the ability of standard community detection algorithms to find intuitively separable neighborhoods. We show that by adding edge weights based on the similarity of localized subgraph features we can apply modularity-based community detection algorithms to uncover intuitively distinct neighborhoods. The use of local network characteristics allow the feature analysis to be completed in linear time, thus making the approach expandable to very large networks. We demonstrate this technique with an application to central Tokyo.

Keywords: Spatial networks · Transportation networks · Community structure · Urban complexity

1 Introduction

When analyzing geospatial data at the mesoscopic level (e.g. regions, counties, metropolitan areas), we may struggle to find a breakdown of a wide area into intuitive and useful regions of analysis. Official administrative boundaries may not exist at the appropriate level and are unlikely to divide the area into natural clusters. Our goal is to use micro level geospatial and network data to divide a large area into its organic neighborhoods. The identification of such neighborhoods is valuable for city planning, pricing models, real estate recommendations, geospatial visualizations, among other applications.

We define neighborhoods as “localities with similar characteristics separated by localities with dissimilar characteristics.” Such a description draws an obvious

parallel with network community structure. However, due to physical constraints and their transportation purposes, road networks rarely exhibit sufficient density and connectivity variation to allow community structure algorithms alone to identify coherent mesoscopic structures. Specifically, community structure typically succeeds in distinguishing areas separated by rivers, highways, or railways, but fails to consistently separate areas with more nuanced differences in road patterns. Our approach is to first generate edge weights for the road network based on the similarity of nodes' local network features, and then use these edge weights to assist the modularity maximization algorithm's ability to cut the network at natural boundaries.

Using this technique, neighborhoods are emergent properties of the road network structure. Clearly the incorporation of population, employment, store, building height, greenery, water, etc. data would help identify perceptually similar and dissimilar areas. However, such data comes in the shape of grids, polygons, or administrative areas that are large with respect to the size of natural neighborhoods and would impose unnatural data gradations at the shape boundaries. Our assumption is that characteristics such as building height, floor area, neighborhood age, and zoning (residential, commercial, or industrial) are sufficiently correlated with features of the road network (length and straightness of edges, proportions of intersection types, etc.) that measures of the road network can indirectly distinguish these perceptual characteristics. For these reasons we focus on road network features that exist at the level of nodes and edges for discovering neighborhoods, with the later integration of rich geospatial data for describing and classifying them.

2 Data Sets

Network Data. Our base network data is the road network for the Tokyo area from Open Street Map (OSM) [12]. The OSM road network includes nodes for all intersections as well as nodes to capture the curvature of the roads with straight edges. The full network from OSM is simplified by merging edges across nodes with degree 2; thus most nodes in the simplified graph correspond to intersections.¹ We limit our analysis to the central 23 wards of Tokyo; an area covering 614 km² [7] that is predominantly urban with a population of 9,172,273 [10] and 7,153,658 jobs [9]. Within this region there are 179,112 nodes and 262,386 edges in the simplified undirected graph. The degree distribution of our road network is presented in Table 1.

Road Characteristics. Roads are tagged by type in the OSM data (e.g. motorway, primary, residential); however, the characteristics (such as speed limit and

¹ Nodes of degree 2 are kept when they are either a road structure change (e.g., surface to tunnel or bridge) or exist at the edge of a network tile. Our network data is segmented into 1500 m × 1500 m tiles to make it manageable in computer memory, and a road segment is kept unmerged if it crosses a tile boundary. This second condition accounts for most of the remaining degree-2 nodes.

Table 1. Degree distribution of nodes in Central Tokyo.

Degree	Count
1	15467
2	17809
3	110414
4	34758
5+	664

road widths) vary within a category and overlap across categories. In the case of Japan, road widths are rarely input into OSM, so we have filled the gaps using the Japanese standard values [6,8]. Because we have simplified the network, the edge geometries are no longer straight lines; they are ‘linestrings’ capturing the concatenated segment lines. In this way, edge straightness is the ratio of Euclidean endpoint distance over linestring length.

3 Analysis Methods

As stated in Sect. 1, our method weighs edges of the road network by the similarity of the nodes’ local network features. For each node in the road network, we collect subgraphs for 3 steps, 5 steps, 155 m and 258 m (the median edge length is 51.5 m, so, 3 steps is \approx 155 m and 5 steps is \approx 258), but this paper focuses on the 5-step case. An edge is included if both end nodes are within the appropriate range.

Variables and Measures. There is a vast literature on analyzing road networks to estimate/predict movement activity using features of the network structure [11,13], optimize logistics [3,4], and perform structural comparisons [1,2]. Naturally, these studies use network measures appropriate for that task (e.g. angular closeness and betweenness centralities). Because our goal here is instead capturing features that can distinguish neighborhoods, a different suite of measures and a novel method to integrate them are necessary.

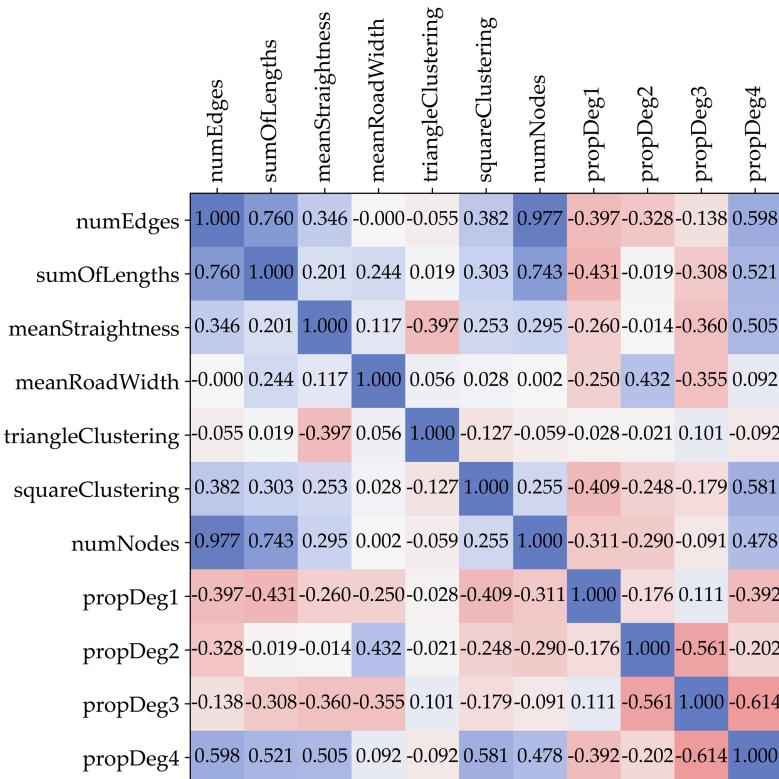
For each ego-centric subgraph we collect the ten variables listed in Table 2.

These measures were chosen both because they capture perceivable features of a road network that may contribute to a neighborhood’s identity and because they are relatively fast to compute (i.e., compared to centrality measures and angularity). Specifically, for each subgraph, all measures can be computed in linear time with a single pass through that subgraph’s edge list.

The level of correlation among these variables is surprising low, excepting the number of nodes, number of edges, and the sum of edge lengths as can be seen in Fig. 1. Note that square clustering (meshedness) is only moderately (0.581) correlated with the proportion of degree four nodes, and even less correlated (0.253) with straightness – both lower than expected.

Table 2. Measures collected for each ego-centric subgraph.

1	Number of edges
2	Total edge lengths
3	Mean edge straightness
4	Mean road width
5	Number of nodes
6	Proportion of degree 1 nodes
7	Proportion of degree 3 nodes
8	Proportion of degree 4 nodes
9	Triangle clustering
10	Square clustering (meshedness)

**Fig. 1.** Pearson correlation matrix for the local network measures using a 5-step subgraph.

Algorithms. Recall our definition of neighborhoods as “regions with similar characteristics separated by areas with dissimilar characteristics.” The unweighted version of the *greedy modularity communities* algorithm, as implemented in

NetworkX [5], finds communities wherein there are more internal connections than external connections. By applying weights calculated as one minus the Euclidean distance of the feature vectors of the standardized variables ($\hat{x}_i = \frac{x_i - \mu}{\sigma}$), the modularity therefore reflects stronger similarity within the community than outside. We experimented with a few different distance metrics (log, truncated, geometric, etc.) and several subsets of the variables, but achieved the best results with linear Euclidean distance on the full set of variables.

4 Results for Central Tokyo

There is no ground truth for organic neighborhoods, but we can evaluate the performance of models using clear perceptual features of the network. For example, grouping together an area with few, large, and straight roads together with an area having many small twisting roads indicates a failure of the model to capture the kinds of neighborhoods we wish to discover. Specifically, based on domain knowledge of the area, we establish three criteria for a proper clustering (visualized in Fig. 2):

1. Marunouchi area is separate from Nihonbashi
2. Nishi Shinkuku is separate from the Shinjuku station area
3. Jingumae is separate from Dogenzaka

First we present the results of using *greedy modularity* without edge weights in Fig. 3. After exploring a range of values, we find that using a modularity resolution parameter (γ) of 4 (less than 1 favors larger communities while greater than 1 favors smaller communities) produces community sizes that capture approximate intuitive community shapes. As expected, unweighted modularity suffices to separate neighborhoods when rivers, large motorways, or train tracks act as barriers, but fails to separate well-connected areas of very different road patterns.

Augmenting the network with similarity weights is expected to allow modularity to find communities of similar network features. However, as shown in Fig. 4, the edge weights do not reveal coherent clusters of similar nodes across large areas of the map. Because each node's features aggregate across 5-step subgraphs, we expected a stronger smoothing effect, but in the dense and heterogeneous road network of central Tokyo, swapping a few roads can (and does) make a large difference in the aggregated subgraphs features. For this reason, some alternative community structure algorithms (e.g. label spreading or Girvan-Newman-style iterative edge cutting) cannot find intuitive neighborhood structures. Because we are using these similarity-based edge weights to augment the modularity maximization, we can still rely on modularity to do the heavy lifting in discovering neighborhood boundaries, while utilizing these weights to modulate exactly where the separations occur.

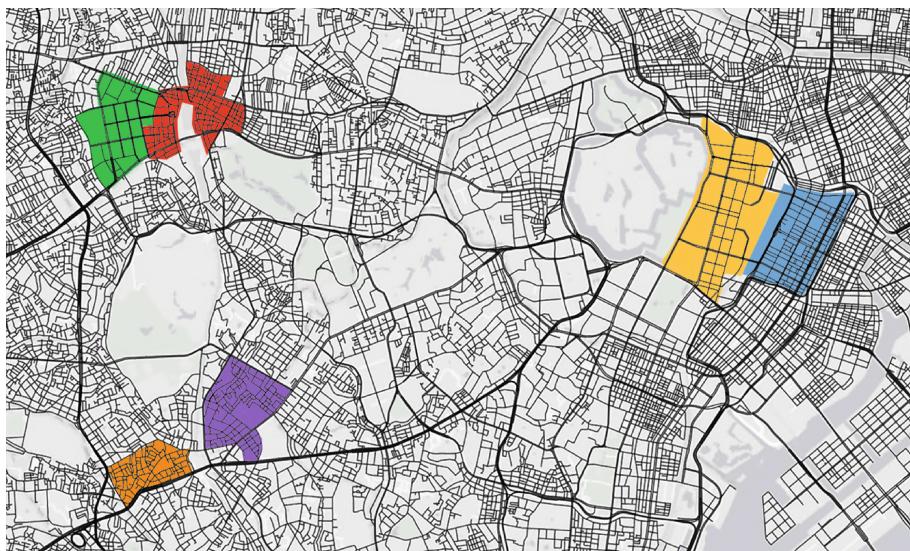


Fig. 2. Visualization of the three pairs of areas used as criteria for a proper clustering. Maurnouchi (yellow) vs Nihonbashi (blue), Nishi-Shinjuku (green) vs Shinkuku Station (red), and Jingumae (purple) vs Dogenzaka (orange). Pictured area is 9.33 km wide by 5.6 km tall.

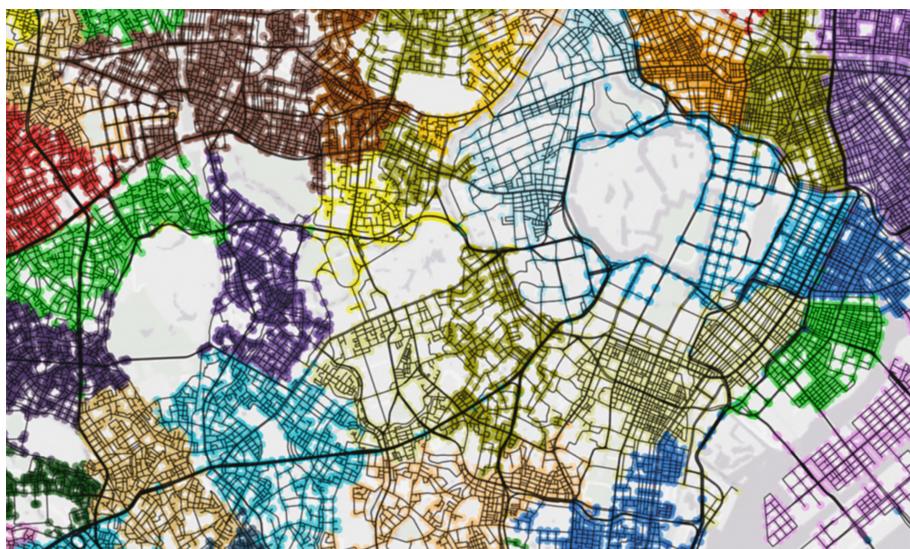


Fig. 3. Close-up map of the communities discovered using the greedy modularity algorithm without edge weights. Criteria 1 and 2 are not satisfied, although the results are reasonable overall.



Fig. 4. Close-up map of the edge weights for node similarity among all variables for 5-step subgraphs. In some areas there is a cohesive neighborhood pattern, but in many areas there is no cohesive pattern.

Across all the subgraph sizes, distance measures, variable sets, and modularity parameters that we analyzed the best results were obtained by the 5-step subgroups using linear Euclidean distances among all variables with a resolution of 4 (detail shown in Fig. 5 and the full 23 wards in Fig. 6). Although not the only combination that satisfies all three criteria, it is one of the few that does. The revealed communities also conform to other intuitive characteristic neighborhoods in the detailed area as well as across the whole 23 wards. For example, the community shapes conform well to unofficially, but popularly identified regions such as the Roppongi Area, Ginza Area, and Shimbashi Area. There are stations and administrative areas with these names, but the intuitive neighborhood bearing that name is better captured by these discovered neighborhoods than the official boundaries.

In some of the regions (especially in the suburbs) the neighborhood separations seem arbitrary; the community size prescribed by the resolution parameter is smaller than the area of similar network structures. However, it is easier to join such communities in post-processing than to modify the modularity algorithm to accommodate larger community size disparities.

Using the same resolution parameter for all the experiments generates a similar, but varying, number of communities as shown in Table 3. Although intuitively we could make more, smaller communities and merge them in

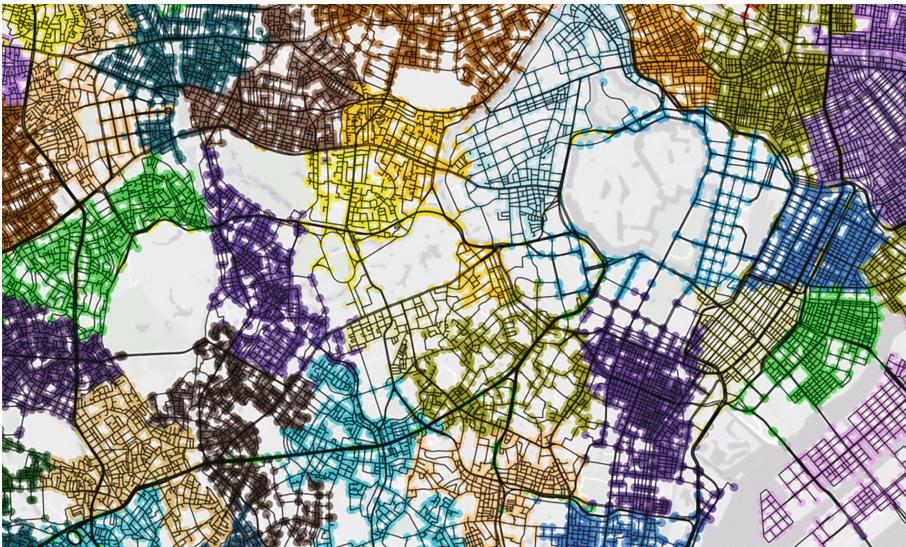


Fig. 5. Close-up map of the communities discovered using the greedy modularity using edge weights based on all variables for 5-step subgraphs. The discovered communities have natural shapes, capture intuitive neighborhoods, and satisfy all three criteria.

Table 3. Number of communities found for select parameter sets.

Parameter set	Number of communities
Base case (no edge weights)	295
5-step, all variable	301
5-step, graph variables	301
258 m, all variables	302

post-processing using other demographic and environmental data to get the results we want, this may not be a viable approach. For other parameter sets the number of communities reached as high as 352, but they performed less well in separating according to our three criteria. Furthermore, generating communities much smaller than these introduces difficulties in resampling and aggregating data available in large grids or administrative areas. That is, it reintroduces the worry that the artificial boundaries of these datasets will bias the shapes of the discovered neighborhoods.

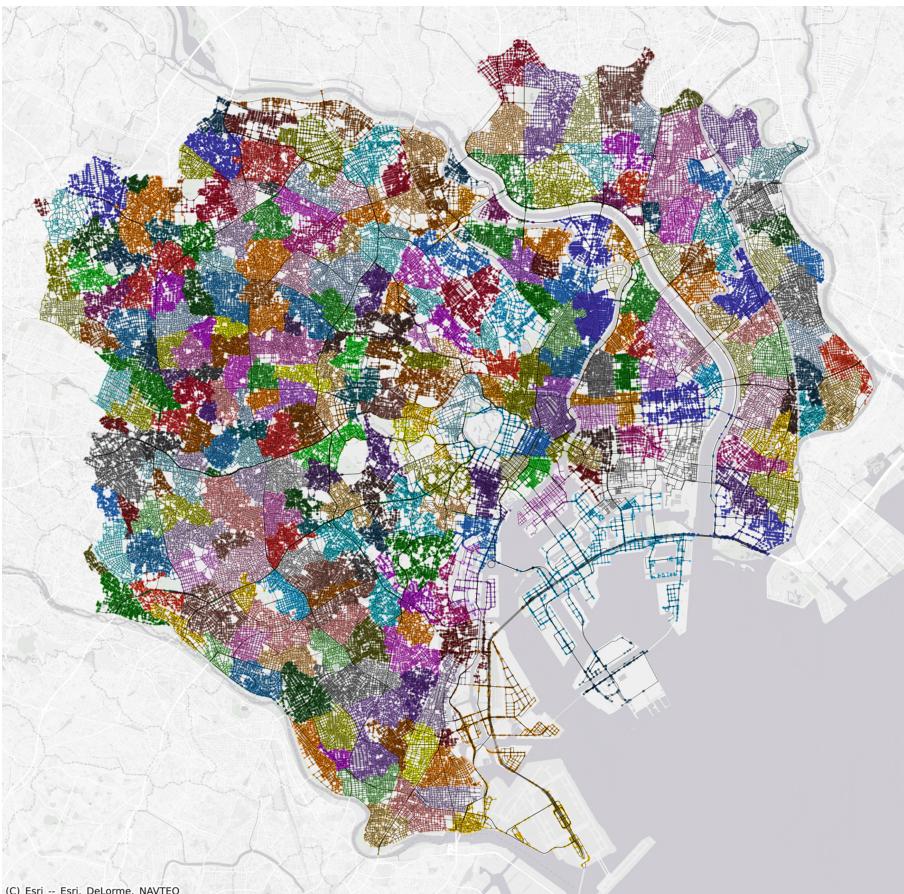


Fig. 6. Discovered neighborhoods for the 23 ward region using all variables on 5-step subgraphs. Note that there are 54 colors to represent 301 communities, so some distinct neighboring group may appear in the same color.

5 Conclusions and Future Work

We have shown that the standard *greedy modularity maximization* algorithm can already find reasonable neighborhoods in parts of Central Tokyo due to typical dividing/connecting features such as large roads, rivers, and railways. However, that alone is insufficient to distinguish dissimilar areas that are nonetheless well-connected. By augmenting the network with edge weights based on the similarity of local network characteristics, we successfully separate dissimilar areas to discover intuitive neighborhoods.

Although the similarity-based edge weights are critical for allowing modularity to find the organic neighborhoods, most sets of variables, subgraph sizes, and distance scalings fail to achieve the desired results. The lack of coherent patterns

in much of the edge weight data reveals that the communities found may not be robust. Alternative edge weighting schemes may provide stronger results.

Other Community Detection Algorithms. We also applied label propagation and the Girvan-Newman method using the maximum edge weight for the iterative removal function. Due to the fragmented nature of the edge weights noted above (Fig. 4) these methods yield tens of thousands of communities, making them ill-suited for our purpose.

The *greedy modularity communities* algorithm we use creates partitions of contiguous nodes into communities. This is a desirable feature for our proof of concept, but we acknowledge that some nodes are interstitial and should not be a member of any community. On the other end, communities may overlap and blend into each other. Using a community detection method based on edge-weighted probabilistic walks can achieve both community gaps and overlaps, but such algorithms seem to require the aforementioned refinement of the edge weights to produce useful results. However, like the current work, perhaps the current edge weights would suffice to modulate the cutting points for such methods to produce useful fuzzy boundaries for neighborhoods.

Integrate Demographic and Environmental Data. As already noted, the current method does not consider demographic or environmental information when discovering neighborhoods. One can imagine a predominantly residential area of large apartment buildings contiguous to an area of large office buildings both with similar road network structures. In this case, the two areas may exhibit distinct neighborhood feels that are indiscernible from the road network alone. Incorporating population, employment, greenery, zoning, and other data into the analysis is targeted for future work, but because this data is only available at much larger scales than the micro-subgraphs used here, we expect it to be used in post-processing (i.e., identifying similar neighborhoods and measuring the heterogeneity/cohesiveness of neighborhoods).

This approach has been largely successful in its task: the unsupervised learning of neighborhoods from local road network features. Successful enough to be useful for the purposes of identifying and visualizing similar organic neighborhoods in Tokyo. Further improvement may be achieved through refinement of the edge weights and/or community detection algorithm, however the problem is well constrained enough that there isn't room for significant improvements beyond the results already obtained for central Tokyo. Different desiderata, different locations, and robustness considerations will drive our efforts towards better methods along these lines.

References

1. Austwick, M.Z., O'Brien, O., Strano, E., Viana, M.: The structure of spatial networks and communities in bicycle sharing systems. PLoS ONE **8**(9), e74685 (2013)
2. Barthélémy, M.: Spatial networks. Phys. Rep. **499**(1–3), 1–101 (2011)

3. Gai, W., Du, Y., Deng, Y.: Multi-objective route planning model and algorithm for emergency management. In: Decision-making Analysis and Optimization Modeling of Emergency Warnings for Major Accidents, pp. 113–150. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-2871-8_5
4. Goczylla, K., Cielatkowski, J.: Optimal routing in a transportation network. *Eur. J. Oper. Res.* **87**(2), 214–222 (1995)
5. Hagberg, A., Swart, P., Chult, S.D.: Exploring network structure, dynamics, and function using networkx. Tech. Rep., Los Alamos National Lab. (LANL), Los Alamos (2008)
6. Japan Traffic Safety Association: Rules of the road (2017)
7. Ministry of Land, Infrastructure, Transport and Tourism: Administrative area data (2020). https://nlftp.mlit.go.jp/ksj/gml/datalist/KsjTmplt-N03-v2_4.html. Accessed 26 Jan 2021
8. Ministry of Land, Infrastructure, Transport and Tourist: Road Structure Ordinance (1970). https://www.mlit.go.jp/road/road_e/q4_standard.html
9. Official Statistics of Japan: Economic census for business frame, tabulation of establishments, results for Japan (2014). www.e-stat.go.jp. Accessed 12 Jan 2020
10. Official Statistics of Japan: Subregional Population by Age and Sex from the 2015 Census (2015). www.e-stat.go.jp. Accessed 11 Nov 2020
11. Omer, I., Kaplan, N., Jiang, B.: Why angular centralities are more suitable for space syntax modeling? In: Proceedings of the 11th International Space Syntax Symposium, Lisbon, Portugal, pp. 3–7 (2017)
12. OpenStreetMap contributors: Planet dump retrieved from <https://planet.osm.org> (2021). www.openstreetmap.org
13. Serra, M., Hillier, B.: Angular and metric distance in road network analysis: a nationwide correlation study. *Comput. Environ. Urban Syst.* **74**, 194–207 (2019)



CityChrone: an Interactive Platform for Transport Network Analysis and Planning in Urban Systems

Indaco Biazzo^(✉)

Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Turin, Italy

indaco.biazzo@polito.it

<http://indacobiazzo.me>

Abstract. Urban systems studies in the last decades have greatly benefited from the digital revolution and the accumulation of a massive amount of data. Extracting useful information from these data calls for new and innovative theoretical and computational approaches. This work presents an open-source, modular, and scalable platform for urban planning and transports network analysis, the CityChrone [citychrone.org]. The platform shows, on interactive maps, measures of performances of public transport in cities. The measures are based on the computation of the travel time distance between a large set of points. Thanks to the high efficiency of the routing algorithm developed, the platform allows users to create new public transports networks and showing the effect on mobility in a small amount of time. A preliminary analysis of the user-generated scenarios is presented. All the source code of the CityChrone platform is open-source, and we employ only open data to ensure the reproducibility of results.

Keywords: Public transport networks · Accessibility measures · Urban systems · Urban planning · Temporal networks

1 Introduction

In 2007, for the first time in human history, the population living in urban areas exceeded the global rural population [26]. Moreover, the urbanization process goes on, and it is expected that in 2050 two-third of the world population will live in cities [26]. People in cities are connected thanks to a large number of processes and different interactive networks. The city is a highly non-linear and out of equilibrium process [19], and problems that institutions, stakeholders, and private citizens will have to face will be more and more related to this high level of complexity. Developing sustainable and efficient citizen mobility and commuting systems is one of the most widespread challenge. Nowadays, thanks to the ICT (Information and Communication Technologies) [39], and the subsequent massive quantities of data accumulated gathered the attention of the scientific community and fostered the emergence of many quantitative studies aiming at

identifying statistical patterns behind the dynamics of humans mobility within or between cities [15, 31, 35, 49], as well as on their infrastructures and services [27, 29, 32, 37, 46–48]. Data about cities and their inhabitants' habits are nowadays collected and available for research and commercial purposes. The information extracted from the statistical analysis of the properties coming from mobility-related data can significantly impact everyday life, helping citizens perform better choices in terms of more environment-friendly mobility solutions, more efficient movements in general, and optimal choice of the place to live. Generally speaking, more precise and easy-to-understand information about the criticality or efficiency of transport services in urban environments is essential at each level of modern society (private citizens and companies, public administrations, and research institutes). This work presents the CityChrone project [www.citychrone.org], an interactive public transport network analysis and planning platform. The analysis is based on accessibility measures. The scientific community defined the concept of *accessibility* several decades ago [21, 22, 33], in order to give a precise quantification of the performance of transportation systems per se and with other aspects of people's lives. Despite its importance, there is not a unique possible definition of accessibility: this could depend on the availability of data or the aim of the researchers performing the analysis [21, 30, 40, 45]. On the other hand, such dispersion increases the difficulty for a straightforward interpretation of accessibility metrics, preventing their operational use by policymakers. Recently, new metrics have been proposed [20] in a general framework aiming to provide a unified point of view in which the *temporal dimension* is at the core. These new metrics are robust and general enough to be applied to different urban systems and different means of transportation. Thanks to a crucial modification to an existing very efficient routing algorithm for public transports, the CityChrone platform can compute those quantities in a fast way for public transports. Users can explore different scenarios of public transports networks in nearly real-time (the time needed to re-compute all the accessibility quantities is less than two minutes for medium-sized cities). The CityChrone platform is an open-source project, and it is published on GitHub [1]. The aim is to involve institutions, companies, and private citizens interested in developing an interactive platform for city knowledge, awareness, and planning. The CityChrone project aims to facilitate the way scientific results are presented, enhanced usability and comprehension through an interactive platform on the web. Moreover, particular care has been devoted to using open data or data freely downloadable to ensure the reproducibility of the results. The work is organized in the following way: In the first section, we describe related works, and in the second section, the routing algorithm. Then in the third section, we present the data and preprocessing procedures. In the fourth section the citychrone platform is presented. In the last section, before the conclusion, a preliminary analysis of the public transport network created by users is shown.

2 Related Works

The literature about accessibility measures is vast and started several decades ago, as stated in the introduction. Despite this vast production of theoretical tools, the computation of these quantities in a real case scenario and in more than one particular case are very few. Moreover, even less interactive platforms have been created to visualize these quantities. One of the first web-based platforms was mapnificien [2]. In this platform, the user can choose a city and visualize isochrones, computed considering displacements with public transports, selecting a starting point on the map. Mapnificien does not show accessibility maps. The isochrones are computed only between stops, and the walking area is only roughly estimated. A platform for computing and visualize accessibility measures is described in [41]. The authors define an accessibility measure and an interactive platform to show several informative layers. The proposed measure is not based on actual travel time but instead on a public transport network's "centrality" measure. Each city stop has a score based on several factors, such as the velocity of transportation, the distance from train stations, and the capacity. This measure has more than five free parameters with no constraints, reducing the measure's universality and transparency. The platform should be accessible for one case study, the Baltimore-Washington DC region, but the URL of the demo is not reachable [3]. No source code is available. Another example of accessibility measure is described in [22]. The authors introduced an accessibility measure that gives four possible values to the public transport stops, from low to very high, based on frequency and means of transportation of public transports routes passing through the considered stop. No source-code or web-based platform was released. Closer to our approach is the "The Metropolitan Chicago Accessibility Explorer" [50]. This platform shows accessibility measures based on isochrones. In the platform [4] the user can choose several different layers showing, for instance, the number of jobs or other services reachable within a given time for each census block of the city. The travel time is computed thanks to the OpenTripPlanner library [cite], an open-source routing library. The travel times are precomputed, and the platform shows the results. The source code was not released. The CityChrone platform, described in this work, has several key differences from all the above platforms. CityChrone uses only open data and standard processes that can be easily applied to every urban system where public transports data are available. Moreover, as far the author knows, it is the only available platform, that thanks to the efficiency of the routing algorithm used, users can build new scenarios of public transports and visualize the effect on accessibility measures after a small amount of time (less than 2 min for medium-size cities like Rome and Boston).

3 Accessibility Quantities

The accessibility quantities considered are presented in [20]. Here we give just a quick review. We want to measure the performance of public transports to

explore the space and connect people in a city. The starting idea is that the isochrones, i.e., the surface at equal time distance t from a starting point p_0 at time t_0 , see Fig. 1, could be used to measure the performance of public transports in cities. More extensive is the area of isochrones larger is the portion of the city that is possible to explore given a time t from the starting point. Based on that, we define two measures: the *velocity score* and the *sociality score*.

The *velocity score*, given a starting point p_0 at time t_0 , measure the velocity of exploring the space around the point considered. This measure can be interpret as the average velocity taken a random directions of displacement given a typical travel time. The precise definition is the following: consider the *covered area*, $A(t, (p_0, t_0))$ of a isochrone at time t starting from p_0 at time t_0 . We define an effective ray $\bar{r}(t, (p_0, t_0))$ as: $\bar{r}(t, (p_0, t_0)) = \sqrt{A(t, (p_0, t_0))}/\pi$. Dividing it by the time t we obtain an effective average velocity: $\bar{v}(t, (P_0, T_0)) = \bar{r}(t, (p_0, t_0))/t$. The effective average velocity is defined for every point p_0 in the map and starting time t_0 . The *velocity score* is obtained averaging over the journey time distribution probability $f(t)$:

$$v(p_0, t_0) = \int_0^\infty v(t, (p_0, t_0))f(t)dt, \quad (1)$$

The journey time distribution probability is the probability distribution of travel time on public transport in the city [35]. Observing the *velocity score* of the cities on CityChrone [citychrone.org] it is clear that the center of the city has a very high-velocity score compared to the suburbs. People living in the center of cities are well served by public transports, having all the directions of displacement allowed, usually also with good and fast public transports, like trains and subways. Instead, in the suburbs, only directions towards the center are well served by public transports, and all the others have poor or no public transports services. The *sociality score* measure instead the possibility of meet people starting from the point p_0 at time t_0 . It measures the number of people reachable from p_0 at time t_0 in a typical daily working trip. The definition is similar to the 1, where instead of the effective velocity $v(t, (p_0, t_0))$, we take the average of the amount of people $pop(t, (p_0, t_0))$ living inside the isochrone at time t starting from the point p_0 at time t_0 . This measure considers, at the same time, the public transports services and the density and distribution of population in the city. For the exact definitions, robustness as well as statistical analysis of these accessibility measures on a large set of cities see [20].

4 Routing Algorithm

Calculating the accessibility measures require to compute travel time distances by public transports between each point in a city. In general, if we consider a grid of points covering a medium-sized city, with reasonable steps, e.g., less than 500 m, the number of points is of order $\propto 10^3$ and the number of journeys to compute is of the order of 10^6 – 10^7 for each city. The availability of efficient routing algorithms is indeed mandatory. Moreover, we want a flexible algorithm

that allows for fast computation of all travel time distances between points when the schedules change, meaning that data preprocessing time should be reduced to the minimum. For road networks, one can compute the driving directions in a millisecond or less at the continental scale, but it is not the case for public transport networks [18]. The approaches and speedup techniques used for road network routing algorithms fail [18], or they are not so effective on public transport networks. In the last years, different approaches, not based on the graph structure of the problem, have emerged in literature where the most promising ones are the RAPTOR algorithm [23] and the CSA Algorithm [24]. Between them, the CSA algorithm seems to be the fastest for the computation of the earliest arrival time [25]. Both algorithms are efficient and fast, with short preprocessing time. Both algorithms have some limitations when considering footpaths to change stops and means of transports, reducing the performance and applicability of those algorithms in urban contexts. The algorithm we devised, the ICSA, is based on the CSA algorithm, but we introduced a crucial modification that allows us to use it in urban systems considering realistic footpaths between stops and means of transport. In the Supplementary Material [5], we describe the CSA algorithm and then the ICSA.

5 Urban Tessellation and Data Preprocessing

Cities have no unique and accepted way to define their area and border. Moreover, large urban systems are composed of several different public transport operators, some of which span their operation well beyond the city's limit, up to national scale. In order to limit the area of analysis, we adopt the definition of cities made by OECD/EU as 'functional economic units' [42]. It uses population density to identify urban cores (city core) and travel-to-work flows to identify the hinterlands whose labor market is highly integrated with the cores (commuting zone). We consider in our analysis only the stops inside both regions and only connections that connect them. In order to have uniform measures of the performance of the public transports over the area, we tessellate the city area by a hexagonal grid with distance center to center of nearby hexagons of 0.4 km. Then we retain only hexagons that have at least one stop reachable in 15 min by walk. For each hexagon in the city, we compute the resident population inside. The data has been gathered through the Eurostat Population Grid [14] for the European cities and the Gridded Population of the world made by the Center for International Earth Science Information Network [34]. This spatial population dataset divides the population into squares with a surface of 1 km^2 , while our tessellation uses hexagons of smaller surfaces ($\sim 0.1 \text{ km}^2$). Hence, we assigned the population in each square to the hexagons overlapping them, proportionally to the fraction of overlapping surfaces. The accessibility measures, based on isochrones, use public transport schedules, streets networks and spatial population distributions in cities. These data are nowadays easily downloadable from the web. Public transport companies usually release their schedules in a uniform way, using the GTFS format [6]. From this file it is possible to extract

the locations of the stops and the connections. The variation of public transports connections does not change significantly in the working days [16, 36], so for each city, we choose a Wednesday in the period of validity of the GTFS file that is not a holiday. We downloaded GTFS files from a repository [7], or directly from the public transports company website. For each hexagon and stop present in a city, we compute the time walking distance between all the stops and hexagons reachable in 15 min by walk. The walking path are computed by an OSRM backend [38] with street graphs taken from OpenStreetMap [43]. The walking speed was set to 5 km/h.

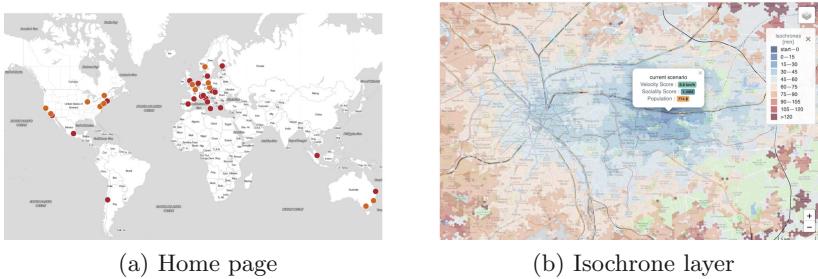


Fig. 1. **a:** home page, where it is possible to select the city to explore accessibility quantity. **b:** Example of isochrone compute in the city of Paris

6 The CityChrone Platform

CityChrone is an open-source web app, developed within the open-source meteor framework [8]. The CityChrone platform is able, given the city's tessellation, the population distribution, the stops, and the connections, to compute isochrones and several accessibility quantities based on public transports and shows them on interactive maps for every city. Moreover, the platform allows users to modify the connections adding new metro lines. After recomputing all accessibility quantities, the user can check how the new scenario changes the accessibility measures.

The User Experience. The platform has three principal sections: the starting page, the visualization page and, the scenario page. On the starting page, Fig. 1a, the user selects the city. The user, clicking on a city, is redirected to the visualization page. In this section, the user can explore different layers that describe distinct aspects of public transport performances. In the left sidebar, the user can select the layers (velocity score, sociality score, population, isochrone and sociality and velocity score difference). The *isochrone layer* shows the isochrones in the city, Fig. 1b. If the user clicks on the map, an isochrone is shown starting from the clicked point. If present, on the left sidebar, it is possible to choose a new scenario with new metro lines added to the city and check the difference

in the isochrones and the accessibility quantities respect the default scenario. In some cities, there is the button “new scenario” by which the user goes to the *scenario page* (Fig. 2).

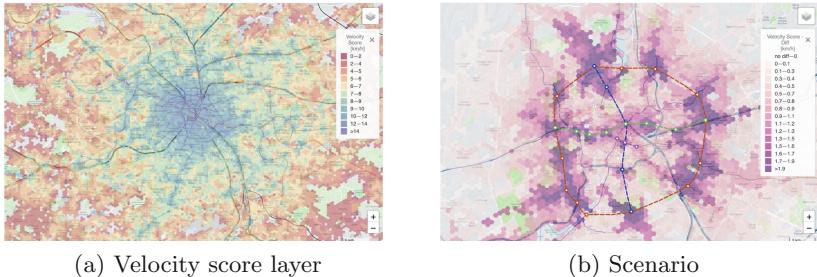


Fig. 2. **a:** Velocity score layer of Paris. **b:** Scenario “rer+circle” create by the user “mat” for the city of Rome with the layer of the improvements in the velocity score.

In the *scenario page* users add new metro lines to the city. There is a limited amount of budget available to users to build new metro lines. The cost of the metro lines is computed given a fixed price for each station and a price per kilometer for the subway tube. The user can add new metro lines by clicking on the “add metro” button. The metro lines added can be dragged, deleted, or expanded (making, for instance, bifurcation). When the constructions of new subways are terminated, the user can click on the “compute” button to compute the new accessibility quantities. In the meanwhile, the user can insert his name and the title for the scenario made. Then, ended the computation, the *visualizator page* is loaded, with the scenario created selected. The page shows the “rank”, highlighting the position of the scenario created. The rank of the scenarios is computed according to the average value of the *velocity score* per person in the city.

Backend. Particular care has been paid to the optimization of some aspect of the platform, the visualization of large amount of data, the minimization of the transfer of data between server and clients, and the scalability of the platform to large audience. The server side of the CityChrone platform is able to, given the data about the public transport of a city, to compute the accessibility quantities and store them in a database. The principal components of the backend are: the data, the routing functions and the accessibility quantities, and the functions related to the visualizations of the information on maps. Working with standard JavaScript objects and HTML code could reduce the performance and the number of objects visualized on the map. Usually, there are about 10^3 of hexagons in a city reaching for Paris the value of $6 * 10^4$. The hexagons are too many to be displayed in a standard browser, so we merge contiguous hexagons with the same color. Empirically we found that their number is reduced by a factor of ten. The most computational demanding operation is the calculation

of accessibility quantities. When a new scenario is created, the computation is made client-side to not overload the server-side. All the routing functions are written to be used both by client and by server-side. On the client-side, the computation exploits the parallel computing that modern browsers allow through Web Workers [9]. The client side computation allows the CityChrone platforms to scale to many users without the need for ample server resources. The current version of CityChrone runs on a virtual server with eight dedicated X86 64bit cores with 16 GB of RAM. The client-side computation of the new scenario required: i) the computation of new connections added to the existing ones, given the list of new metro lines created by users. We assume a fast metro line (see for example [10]), with acceleration of 1.3 m/s^2 , a maximum velocity of 30 m/s and frequency every 2 min. These parameters can easily changed to reflect local project constraints. ii) For each new stop, the walking times with stops and hexagons reachable in 15 min by walk is computed. A OSRM server [11,38] with the street network taken from OpenStreetMap [43] of the city is active on the server. To save server-side resources, each new stop can take the walking paths equal to those of the nearest hexagon or stop (this could reduce the precision of the calculation, but it avoids having an OSRM server always up). We implement both solutions. iii) The computation of the new travel times between all points and the relative accessibility quantities. The running time for the complete computation of a new scenario is about 1–2 min for a medium-size city (Rome, Boston, etc.) on a computer with a 4-core CPU. For a larger city, the time can reach 30 min, and for these cities, we do not allow, so far, users to create a new scenario.

7 Preliminary Analysis of New Public Transport Network Scenarios: The Case of Rome

In this section, we analysed the scenarios created for the city of Rome. In two events, in 2017 and 2018, the platform was presented inside two public scientific events organized in Rome [12]. In the period 2017–2021, the web page of Rome has had about 3000 unique views, and users created more than 350 new public transport scenarios. The users have a budget of 5000M€ of euros to construct new metro lines. We assume that a single stops cost 100M€ (see for instance the average costs of a Paris metro station [13]) and the tube 30M€ per km (average cost of a tunnel boring machine tube [17]). These costs are assumed valid on average for a medium size city, but locally can vary depending on several different factor, ranging from architectural projects to soil composition. More specific cost functions can be easily integrated in the platform. The goal is to maximize the gains in velocity scores or sociality scores of the whole city. In order to analyze the user solutions so far created, we cluster them according to the impact on the accessibility measures considered, the sociality score. We associate to each scenario an array containing the sociality score computed for each point of the hexagonal grid covering Rome, and we cluster the scenarios according to them. We use the affinity propagation algorithm [28] implemented

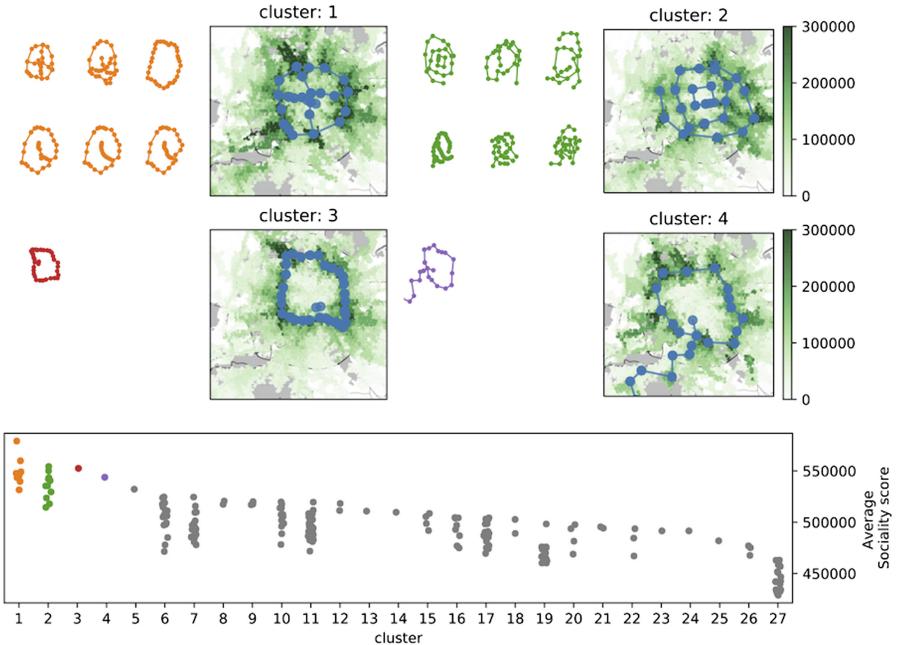


Fig. 3. The figure shows the scenarios of new subways generated by users in the case of Rome. The solutions are clustered according to the distribution on the city of the sociality score. The average value of the sociality score of the scenarios in each cluster is shown in the bottom plot. The clusters are ordered according to the best solution they contain. The four plots above show up to the six best scenarios for the four best clusters. For each cluster, the improvements in the sociality score of the best scenario are shown in the right plots.

in scikit-learn [44] because it does not need to specify the number of clusters beforehand. We rank each cluster of scenarios according to the best scenario it contains. Figure 3 shows the best four clusters, together with the six best metro lines proposed by users (if present). All the four clusters of solutions share a future, the need for a circular line in Rome. The best solution so far seems to merge a circular line with high-speed connections that irradiate from Rome's city center. Cluster 1 shows solutions with a circular line and, usually, some small metro lines enter the city's center. The scenarios belonging to cluster 2 have a more complex shape, with lots of metro stations serving the city center and partial circular lines. Cluster 3 has only one scenario, which is a circular line with very dense stops. The solution of Cluster 4 has a circular line with a fast connection to a west part of Rome that overlooks the sea. Closer inspections of the scenarios are possible on citychrone.org.

8 Conclusion

In this work, we presented CityChrone: an interactive platform for public transport analysis and planning. The platform shows several accessibility measures about the performance of public transports in more than 30 cities around the world. The accessible measures shown are based on the computations of a vast number of travel time distances between a grid of points in the city. The platform exploits a very efficient routing algorithm that allows a fast computation of the accessible quantities. Users can explore new public transports scenarios, build new metro lines, and check quickly (in minutes) how the proposed solutions change the accessibility measures. The primary computational requests are designed to run client-side, allowing the CityChrone platform to scale to a large number of users with small server-side resources. New cities can be easily added to the platform having just the schedules of the public transports. A script process the data, and then, after loaded in the CityChrone platform, it is possible to visualize isochrones, accessibility quantities and test new transports scenarios. The CityChrone platform is the first step of an open-source project aiming to create a community of companies, public institutions, stakeholders, developers, and private citizens interested in developing interactive platforms to analyze the transports in cities and search for innovative solutions to mobility problems in cities.

Acknowledgement. This work has been supported by the SmartData@PoliTO center on Big Data and Data Science.

References

1. <https://github.com/CityChrone>
2. <https://www.mapnificent.net/>
3. <http://www.geosimulation.org/transit.html>
4. <http://urbanaccessibility.com/>
5. <https://bit.ly/3hkG613>
6. <https://developers.google.com/transit/gtfs>
7. www.transitfeeds.com
8. <https://www.meteor.com>
9. <https://html.spec.whatwg.org/multipage/workers.html>
10. <https://tinyurl.com/yyccnsau6>
11. <https://github.com/Project-OSRM/osrm-backend>
12. <http://www.kreyon.net/kreyonDays/>
13. <https://marroninstitute.nyu.edu/events/how-much-does-it-cost-to-build-a-subway>
14. Eurostat population grid. <http://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/population-distribution-demography/geostat>
15. Alessandretti, L., Aslak, U., Lehmann, S.: The scales of human mobility. *Nature* **587**, 402–407 (2020)
16. Alessandretti, L., Karsai, M., Gauvin, L.: User-based representation of time-resolved multimodal public transportation networks. *Royal Soc. Open Sci.* **3**(7) (2016). <http://rsos.royalsocietypublishing.org/content/3/7/160156>

17. Attinà, M., et al.: Annex 13 - case study on tunnels of “assessment of unit costs (standard prices) of rail projects (capital expenditure)” (2018). <https://tinyurl.com/48chu6r8>
18. Bast, H., et al.: Route planning in transportation networks. In: Kliemann, L., Sanders, P. (eds.) Algorithm Engineering. LNCS, pp. 19–80. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49487-6_2
19. Batty, M.: The size, scale, and shape of cities. *Science* **319**(5864), 769–771 (2008)
20. Biazzo, I., Monechi, B., Loreto, V.: General scores for accessibility and inequality measures in urban areas. *Royal Soc. Open Sci.* **6**(8), 190979 (2019)
21. Black, J., Conroy, M.: Accessibility measures and the social evaluation of urban structure. *Environ. Plan A* **9**(9), 1013–1031 (1977)
22. Bok, J., Kwon, Y.: Comparable measures of accessibility to public transport using the general transit feed specification. *Sustainability* **8**(3), 224 (2016)
23. Delling, D., Pajor, T., Werneck, R.F.: Round-based public transit routing. *Transp. Sci.* **49**(3), 591–604 (2014)
24. Dibbelt, J., Pajor, T., Strasser, B., Wagner, D.: Intriguingly simple and fast transit routing. In: Bonifaci, V., Demetrescu, C., Marchetti-Spaccamela, A. (eds.) Experimental Algorithms SEA 2013. LNCS, vol. 7933, pp. 43–54. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38527-8_6
25. Dibbelt, J., Pajor, T., Strasser, B., Wagner, D.: Connection scan algorithm. CoRR abs/1703.05997 (2017). <http://arxiv.org/abs/1703.05997>
26. Department of Economic and Social Affairs, Population Division: World urbanization prospects: The 2014 revision, highlights (ST/ESA/SER.A/352). Population Division, United Nations (2014)
27. Fleurquin, P., Ramasco, J.J., Eguíluz, V.M.: Characterization of delay propagation in the US air-transportation network. *Transp. J.* **53**(3), 330–344 (2014)
28. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. *Science* **315**(5814), 972–976 (2007)
29. Gallotti, R., Bazzani, A., Rambaldi, S.: Understanding the variability of daily travel-time expenditures using GPS trajectory data. *EPJ Data Sci.* **4**(1), 18 (2015)
30. Geurs, K.T., Van Wee, B.: Accessibility evaluation of land-use and transport strategies: review and research directions. *J. Transp. Geogr.* **12**(2), 127–140 (2004)
31. Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.L.: Understanding individual human mobility patterns. *Nature* **453**(7196), 779–782 (2008)
32. Guimera, R., Mossa, S., Turtschi, A., Amaral, L.N.: The worldwide air transportation network: anomalous centrality, community structure, and cities’ global roles. *Proc. Natl. Acad. Sci.* **102**(22), 7794–7799 (2005)
33. Hansen, W.G.: How accessibility shapes land use. *J. Am. Inst. Plann.* **25**(2), 73–76 (1959)
34. Center for International Earth Science Information Network CIESIN Columbia University: Gridded Population of the World, Version 4 (GPWv4): Population Count (2016). <http://dx.doi.org/10.7927/H4X63JVC>
35. Kölbl, R., Helbing, D.: Energy laws in human travel behaviour. *New J. Phys.* **5**(1), 48 (2003)
36. Kujala, R., Weckström, C., Darst, R.K., Mladenović, M.N., Saramäki, J.: A collection of public transport network data sets for 25 cities. *Sci. Data* **5**, 180089 (2018)
37. Li, R., et al.: Simple spatial scaling rules behind complex cities. *Nat. Commun.* **8**(1), 1–7 (2017)

38. Luxen, D., Vetter, C.: Real-time routing with OpenStreetMap data. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS 2011, pp. 513–516. ACM, New York (2011). <https://doi.org/10.1145/2093973.2094062>
39. Mayer-Schönberger, V., Cukier, K.: Big Data: A Revolution that Will Transform How We Live, Work, and Think. Houghton Mifflin Harcourt, Boston (2013)
40. Miller, H.J.: Measuring space-time accessibility benefits within transportation networks: basic theory and computational procedures. *Geogr. Anal.* **31**(1), 1–26 (1999)
41. Mishra, S., Welch, T.F., Torrens, P.M., Fu, C., Zhu, H., Knaap, E.: A tool for measuring and visualizing connectivity of transit stop, route and transfer center in a multimodal transportation network. *Public Transp.* **7**(1), 77–99 (2014)
42. OECD: Redefining “Urban”. OECD Publishing (2012). <https://www.oecd-ilibrary.org/content/publication/9789264174108-en>
43. OpenStreetMap contributors: Planet dump (2017). <https://planet.osm.org> <https://www.openstreetmap.org>
44. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
45. Schmöcker, J.D., Quddus, M.A., Noland, R.B., Bell, M.G.: Mode choice of older and disabled people: a case study of shopping trips in London. *J. Transp. Geogr.* **16**(4), 257–267 (2008)
46. Sen, P., Dasgupta, S., Chatterjee, A., Sreeram, P., Mukherjee, G., Manna, S.: Small-world properties of the Indian railway network. *Phys. Rev. E* **67**(3), 036106 (2003)
47. Sun, X., Wandelt, S., Hansen, M.: Airport road access at planet scale using population grid and OpenStreetMap. *Netw. Spat. Econ.* **20**(1), 273–299 (2020)
48. Sun, X., Wandelt, S., Zhang, A.: Comparative accessibility of Chinese airports and high-speed railway stations: a high-resolution, yet scalable framework based on open data. *J. Air Transp. Manag.* **92**, 102014 (2021)
49. Yan, X.Y., Zhao, C., Fan, Y., Di, Z., Wang, W.X.: Universal predictability of mobility patterns in cities. *J. R. Soc. Interface* **11**(100), 20140834 (2014)
50. Yin, S., Li, M., Tilahun, N., Forbes, A., Johnson, A.: Understanding transportation accessibility of metropolitan Chicago through interactive visualization. In: Proceedings of the 1st International ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics, pp. 77–84. ACM (2015)



VLT-LUT: Modeling the Very Long-Term Evolution of the City in 300 Years

Ariel Castillo¹, Francisco Martínez², Pedro Donoso¹, Leonel Gutiérrez¹,
and Ricardo de la Paz Guala¹(✉)

¹ Universidad de Chile, Santiago, Chile

{ariel.castillo,pedro.donoso,lgutierrez}@ing.uchile.cl,
ricardo.delapaz@uchile.cl

² Institute of Complex Engineering Systems (ISCI), Universidad de Chile,
Santiago, Chile
fmartine@u.uchile.cl

Abstract. The evolution of cities is modelled by developing a software to simulate the effects of agglomeration economies and transportation planning in the very long term. The cities' complexity is modeled with VLT-LUT, a tool that simulates the evolution of land use and transportation and their interaction, in a time span of 300 years, or from 1 to 29 million inhabitants, based on urban microeconomic theory and market equilibrium. Preliminary simulation results on an artificial city include the impact of exogenous scenarios of road network evolution and of the agents' perception of agglomeration economies, observing the evolution of land-use forms and the city size. Another result is that land rents evolve super-linearly with population, in line with previous empirical and theoretical research, but its strength is differentiated by scenario.

Keywords: Land use · Transportation · Urban models · Urban networks · Cities scaling

1 Introduction

In recent decades, complex models of cities have been formulated and applied worldwide, especially for urban policies analysis, generically named *land use and transportation (LUT)* models or LUTE when it includes the production and labor markets (reviewed in [6]). Such models share the approach of the interaction between land use and transportation models, transferring information of transportation costs and location of activities. Their aim is to simulate city functions under different policy scenarios, usually in a time span of 20 to 30 years, through the interaction between representative household and firm agents and the movement of goods, providing different performance indices that allow the assessment of policy scenarios.

This medium-term perspective is useful for planning but lacks an understanding of the very long-term evolution. In this wider perspective, there is evidence

of universal laws that emerge from worldwide urban big data [9,11,12], similar to those observed in nature's complex systems [10]. These are scaling laws with profound implications in the evolution of urban systems: they are superlinear (output per capita increases with population) i.e., despite differences in history (culture or geographical), [10] claims that cities follow a universal law (power law) with a common scale parameter bounded to ± 0.15 . In contrast to this evidence, superlinearity is not a feature of LUT models, as they are designed as sets of microeconomic rules econometrically adjusted to each context. To explain the scaling law, LUTE [4] and CLUTE [5] microeconomic models were formulated allowing the interaction between cities through demographic migration (geographically and socio-economically). CLUTE explains how the scaling law emerges from the complex interaction in the urban system, particularly, from individuals' and firms' rationality under uncertain information. With this theoretical foundation, the emergence of an urban science is conceived [5].

This paper reports—to our knowledge—the first attempt to simulate the evolution of cities in the very long term: a period of about 300 years or from 1 to 29 million inhabitants. It is designed as a research platform to analyze the different paths that a city may take in the very long term according to its agglomeration economies and transportation network, to extract lessons regarding the quality of life, productivity, and sustainability of the environment. As a proof of concept, we present a prototype model, the *very long-term land use and transportation (VLT-LUT)* software where economic interactions are replaced by gross accessibility indices. We report preliminary results of the evolution of a fictitious city, simulated in a symmetric, flat, and homogenous plain, initialized with population and firms distributed homogeneously in the space and run for exogenous increments of population representing time steps, keeping memory of the past built infrastructure that makes the model dependent on the steps chosen.

The aim of this paper, at this stage of our research, is not to replicate the evolution of a real city but to evaluate the performance of the VLT-LUT model under controlled scenarios. Given the symmetric conditions of the geography and the transportation network, we focused on analyzing whether the form of the complex urban system remains symmetric in the very long-term simulation and whether the scaling law is replicated. The reported results show a symmetric evolution with different outcomes according to the behavior of agents regarding two main factors: accessibility and agglomeration economies. We remark that values for factors and parameters that are used in this paper have been conveniently chosen, in order to observe the performance of the model while keeping the simplicity required for this analysis.

2 Very Long-Term Land Use and Transportation Model

In this section, we present the core contribution of this paper, the *very long-term land use and transport (VLT-LUT)* model, summarizing each submodel and setting some parameters. VLT-LUT represents the long-term evolution of an artificial city, considering a private transport network in a homogenous discrete land divided into 400 zones and the growth of agents, with residents categorized by

socioeconomic attributes and firms by economic sector. The model is initialized inputting homogeneous allocation, including zone and building type, of 1 million agents in a city with a given initial size and a given road network.

2.1 Land Use Model: CUBE Land

Agent's demand is modeled by calculating agent h 's probability to be allocated in zone i and real estate type v , based on agents' bids for each location, computed as $\hat{B}_{hvi} = B_{hvi} + \xi$, where B_{hvi} is the deterministic component and ξ is a Gumbel error. The deterministic part is given by $B_{hvi} = b_h + b_{hvi} + b$, where: b_h is a reference bid that adjusts the utility levels to reach location equilibrium for each type of agent h (solved according to FPP3 below); b is a constant that adjusts bid levels to absolute price values in the economy; b_{hvi} is agents h 's value of zone i and real estate v . Because b_{hvi} includes the set of attributes describing neighbours' location externalities and economies of agglomeration, i.e., $b_{hvi}(P_{\cdot|i}, S_{\cdot i})$, then the demand model solves fixed point problem, FPP1 on auction allocation $P_{h|vi} = f(P_{\cdot|i})$:

$$P_{h|vi} = \frac{H_h \exp(\mu(b_h + b_{hvi}(P_{\cdot|i}, S_{\cdot i})))}{\sum_g H_g \exp(\mu(b_g + b_{gvi}(P_{\cdot|i}, S_{\cdot i})))}, \quad (\text{FPP1})$$

where $P_{h|vi}$ maximizes the stochastic bids simulating an auction, H_h is the total of agents of type h and S_{vi} is the supply of real estate type v in each zone i .

Rents r_{vi} result from the auction process and acquire the value of the maximum willingness to pay at each location and, given the Gumbel distribution of the bids errors, are represented by (1) (endogenously supply-dependent, given construction's economy of scale):

$$r_{vi} = \frac{1}{\mu} \ln \left(\sum_h H_h \exp(\mu B_{hvi}) \right). \quad (1)$$

The land use model predicts the real estate market, estimating the supply S_{vi} for different types of real estate properties v in each zone i , for a situation with partial demolition of previous supply and based on a profit maximization Logit model. The model solves fixed point problem FPP2 on supply $S_{vi} = f(S_{\cdot \cdot})$:

$$S_{vi} = S_{0vi} (1 - k_{vi}) + (S - S_0 - S_D) \frac{\exp(\lambda \pi_{vi}(S_{\cdot i}))}{\sum_{w,j} \exp(\lambda \pi_{wj}(S_{\cdot j}))}, \quad (\text{FPP2})$$

with $S = \sum_{vi} S_{vi} = \sum_h H_h$, $S_0 = \sum_{vi} S_{0vi}$, and $S_D = \sum_{vi} S_{0vi} k_{vi}$, where: S_{0vi} is the supply corresponding to the previous population level; k_{vi} is the demolition rate; π_{vi} is the expected value of developer profits $\hat{\pi}_{vi}$, defined as $\hat{\pi}_{vi} = r_{vi} - C_{vi} + \varepsilon$, where cost functions C_{vi} are given by Eq. (5), and ε is a Gumbel error.

The equilibrium between demand and supply models is reached assuming that all agents are allocated, which is attained by adjusting b_h variables of bids,

solving the demand-supply equation $H_h = \sum_{vi} S_{vi} P_{h|vi}$, where $P_{h|vi}(b_h)$, i.e., the equilibrium is a fixed point problem on b_h for all h ([FPP3](#)). CUBE Land model solves FFP3 and the result is the location matrix $H_{hvi} = S_{vi} P_{h|vi}$.

$$b_h = -\frac{1}{\mu} \ln \left(\sum_{vi} S_{vi} \exp(\mu(b_{hvi} - r_{vi})) \right). \quad (\text{FPP3})$$

Grid and Variables. The grid defines 400 square zones of 25 km². Each zone i , $i = 1, \dots, 400$ represents a location option for any of the 5 types of agents, residential and non-residential, to locate any of the 5 types of housing (see [Table 1](#)).

Table 1. Types of agents and types of properties

Agent h	Characteristic	Property v	Characteristic
1	Low-income home	1	Small house
2	Mid-income home	2	Big house with yard
3	High-income home	3	Apartment/Office
4	Industry	4	Commercial store
5	Commerce	5	Large lot

We define three types of attributes:

- Agents' attributes: The agents' variable is the income I_h , which is exogenous and defined only for residential agents ($h = 1, 2, 3$), with $I_1 = 13.283$ UF, $I_2 = 25.618$ UF, and $I_3 = 37.953$ UF (UF is a Chilean currency).
- Housing attributes: For each type of housing $v = 1, \dots, 5$, we define two attributes: building size q_c^v and land size q_t^v . The first one represents the size of the construction plan of housing type v and the second represents the land lot size used by v . The values assumed for these attributes are shown in [Table 2](#), where: $v = 1$ is a back-to-back house; $v = 2$ is a detached house; $v = 3$ is a flat; $v = 4$ is an office; $v = 5$ is an industrial property.

Table 2. Attributes of each type of property v (m²)

v	1	2	3	4	5
q_c^v	35	70	50	50	300
q_t^v	35	140	1	50	500

- Zonal attributes: We define the residential, industrial, and commercial densities of zone i , ρ_i^{res} , ρ_i^{ind} , and ρ_i^{com} (hab/km²), respectively, as:

$$\rho_i^{res} = \sum_{h=1}^3 \sum_{v=1}^5 H_{hvi} \frac{\theta_h}{A_i}, \quad \rho_i^{ind} = \sum_{v=1}^5 H_{4vi} \frac{1}{A_i}, \quad \text{and} \quad \rho_i^{com} = \sum_{v=1}^5 H_{5vi} \frac{1}{A_i}, \quad (2)$$

where A_i is the zone area (25 km^2), and θ_h is the average number of inhabitants by type of agent h . We use $\theta_h = 3.5$ inhab/house for $h = 1, 2, 3$. Densities represent a type of location externalities because they depend on the location of agents other than the bidder, thus introducing in the agent's bid B_{hvi} the location of others, i.e., bids depend on the location probability of all other agents and it is an endogenous variable in CUBE Land.

Access is also an endogenous variable in the LUT model, as it depends on densities and transportation costs. For each zone i , access is described by two components, *accessibility* to commercial activities (acc_i) and *attractiveness* to residents (att_i), defined as follows:

$$acc_i = \ln \left(\sum_{j=1, j \neq i}^{400} (\rho_i^{ind} + \rho_i^{com}) \exp(-\alpha_0 \tau_{ij}) \right), \quad (3)$$

$$att_i = \ln \left(\sum_{j=1, j \neq i}^{400} \rho_i^{res} \exp(-\alpha_0 \tau_{ji}) \right), \quad (4)$$

where, α_0 is a parameter equal to 0.03 min^{-1} , defining the disutility of τ_{ij} , assumed, while τ_{ij} represents the expected minimum travel time between the centroids of zones i and j (see Subsect. 2.2).

Cost Functions. The cost function of housing type v in zone i , C_{vi} , is the construction cost plus the land lot costs, thus, we have that

$$C_{vi} = \alpha_1 q_c^v + p_i q_t^v, \quad (5)$$

where $\alpha_1 = 0.009 \text{ UF/m}^2$. Land prices p_i (UF/m^2) are endogenous variables that depend on rents, while rents depend on supply S_{vi} , supply depends on the profits π_{vi} , and these depend on costs C_{vi} which depends on p_i , then $p_i = f(p_i)$ and can be expressed by fixed point problem FPP4.

$$p_i = \min_w p_{wi} = \min_w \frac{r_{wi}(p_i)}{q_t^v}. \quad (\text{FPP4})$$

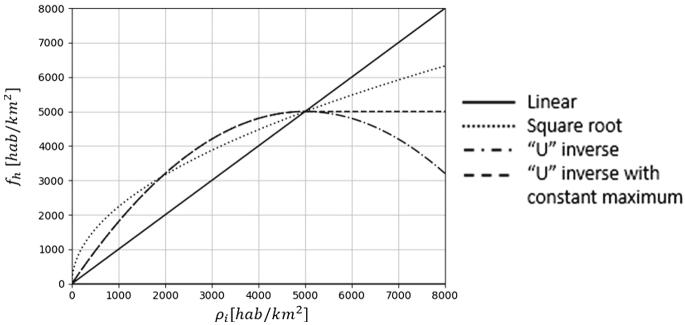
Bid Functions. We define agent h 's bid for property type v in zone i , b_{hvi} , as:

$$b_{hvi} = \begin{cases} \alpha_h I_h + \alpha_{1h} acc_i + \alpha_{2h} f_h(\rho_i^{res}) + \alpha_{3h} q_c^v + \alpha_{4h} q_t^v, & \text{if } h = 1, 2, 3, \\ \alpha_{1h} att_i + \alpha_{2h} \rho_i^{ind} + \alpha_{3h} q_c^v + \alpha_{4h} q_t^v, & \text{if } h = 4, \\ \alpha_{1h} att_i + \alpha_{2h} \rho_i^{com} + \alpha_{3h} q_c^v + \alpha_{4h} q_t^v, & \text{if } h = 5, \end{cases} \quad (6)$$

where values of α_h parameters are shown in Table 3. We also consider that the supply and demand Logit probabilities have scale parameters $\lambda = 0.03$ and $\mu = 2.5$, respectively. f_h represents the agent's perception of the density attribute, for which we consider four different shapes (Fig. 1), where the base scenario is a linear function.

Table 3. Parameters used in bids functions

h	α_h	α_{1h}	α_{2h}	α_{3h}	α_{4h}
1	0.1	3.0	0.003	0.002	0.002
2	0.15	3.5	0.001	0.003	0.003
3	0.2	4.5	0.0005	0.004	0.004
4	—	3.0	0.01	0.01	0.05
5	—	4.5	0.002	0.05	0.01

**Fig. 1.** Different density perception functions

2.2 Transportation Model: Markovian Traffic Equilibrium

We first define the transportation network structure. To illustrate how, consider Fig. 2 that depicts the 400 zones grid and a symmetric transport network (1 lane per direction roads) with a central cross-shaped highway (2 lanes per direction). As shown in the inset of Fig. 2, at each zone we define 17 nodes: a centroid (to receive and generate trips), 12 external (to interact with other zones); 4 internal (to interact with the centroid); 28 bidirectional arcs: 4 imaginary that connect internal nodes to the centroid; 24 real connecting non-centroid nodes, that can be of 1 or 2 lines.

We apply the Markovian traffic equilibrium (MTE) [2], a traffic assignment model for private transportation where trips result from recursive arc choices. At a given node n and for a given destination node d , MTE assigns a portion P_a^d of the aggregated flow from all origins arriving at node n going to destination d , x_n^d , among each outgoing arc $a = (n, m) \in A_n^+$ (set of arcs leaving node n), by applying a Logit rule (with fixed dispersion parameter $\delta = 1$) whose criterion is that travelers choose the arc with the expected minimum cost of going from n to d by using arc a , denoted as z_{ad} . Thus, we have that:

$$P_a^d = \frac{\exp(-\delta z_{ad})}{\sum_{b \in A_n^+} \exp(-\delta z_{bd})}. \quad (7)$$

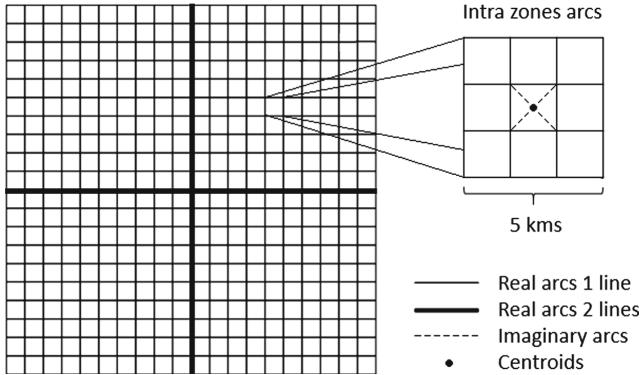


Fig. 2. Structure of the network grid and zones

Here, z_{ad} comes from the sum of the cost of arc a , t_a , and the expected minimum cost from m (a 's end node) to destination node d , τ_{md} , thus $z_{ad} = t_a + \tau_{md}$. Note that expected minimum costs depend on the remaining trip to the destinations, i.e., $z_{ad}(\tau_{..})$, then, the model solves fixed point problem **FPP5** on τ_{md} :

$$\tau_{md} = -\frac{1}{\delta} \ln \left(\sum_{a \in A_m^+} \exp(-\delta z_{ad}(\tau_{..})) \right). \quad (\text{FPP5})$$

On the other hand, the cost of arc a , t_a , depends on its flow w_a :

$$t_a = \begin{cases} t_a^0 \left(1 + b_a \left(\frac{w_a}{c_a}\right)^{p_a}\right), & \text{if } w_a \leq c_a \\ 30[\text{min}], & \text{otherwise,} \end{cases} \quad (8)$$

where: t_a^0 is the free-flow travel time; c_a is the capacity; $p_a = 3$ is a fixed non-linear parameter assumed $p_a = 3$; b_a is a known parameter (values in Table 4). Variable w_a comes from the aggregation of flows of arc $a = (i, j)$ going to each destination d , denoted as v_a^d and computed as $v_a^d = x_i^d P_a^d$, thus, $w_a = \sum_d v_a^d$.

Table 4. Values for the congestion function parameters for every type of arc

Type of arc a	t_a^0 (min)	b_a (adimensional)	c_a (veh/h)
Imaginary	0.1	0	1
Real: bidirectional-1 lane	3.33	2	2000
Real: bidirectional-2 lanes	2	4	4000

2.3 Demographic and Firmographic Models

The demographic and firmographic models estimate the population growth, number of agents, firms, and households. We replicate SECTRA's methodology [7,8]

to estimate the number of agents of each type h , H_h , $h = 1, \dots, 5$, for each forecasting year, based on the national estimate of GDP's growth per capita for mid-sized and large cities.

The demographic model estimates the total number of residential agents (types $h = 1, 2, 3$). For large cities, an estimation of income by decile is first computed, according to GDP, then, the probabilities of being on each decile range are computed to estimate each H_h . On the other hand, for mid-sized cities, the mean income of all agents is used to estimate H_h .

The firmographic model estimates the total number of firms (non-residential agents, types $h = 4, 5$). First, an estimation of the built surface of all types of use (industry, commerce, education, services, and others) is computed. With these results, H_h are estimated considering a linear dependency on the built area, with parameters that differ between large and mid-sized cities.

2.4 Trip Generation and Distribution Model

The land use model's output provides the allocation of agents in the city, which is used to estimate trips between zones using private transportation costs provided by the MTE model.

Given the land use, the trip generation and distribution (GDT) model seeks to estimate the daily movement of inhabitants. We use a standard doubly constrained entropy model to estimate trips between zones by solving a fixed point problem (FPP6), as in [3]. It uses as input the location matrix, the zonal densities, and the last computed travel times to deliver as output the number of trips between all pairs of zones. This obtained demand serves as an input for the MTE model, which then updates the travel times.

3 Simulation Process

We integrate CUBE Land, MTE, and GDT connected by input-output file of access. In general terms, the equilibrium of the LUT model is attained by solving fixed problems FPP1, FPP2, FPP3, and FPP4 in CUBE Land, FPP5 in the MTE transportation model, and FPP6 in the GDT model (Fig. 3).

3.1 Modeling the Urban Spatial Border

As the population grows the city sprawls and the city border has to be modeled. We assume that urban and rural agents compete for location in all 400 zones. This competition is modelled external to CUBE Land and, for each i , according to the auction, with the profit $\pi_i = \max_v (r_{vi} - C_{vi})$, and an exogenous agricultural rent R_A . There are two cases: if the profit is less than R_A , then i is assigned to agriculture, no homes or industries will locate there and it is considered out of the urban boundaries; otherwise, i is included in the set of urban zones.

This process allows agricultural zones to become urban, but the inverse is ruled out, thus avoiding rural zones within the urban limit. This allows the city to extend its urban limits accordingly with the growing population following [1]'s rule to define the city boundary as an auction between rural and urban land use.

3.2 Solution Algorithm

We propose a solution algorithm that can be summarized as follows:

- **Initialization: Iteration 0:** A population of $N = 1$ million is set and distributed by agent type, using the demographic model. Next, an exogenous and homogeneous distribution of agents among zones is applied to compute initial values for the variables using free-flow travel times. The urban limit feature is applied to obtain the Iteration 0 of the algorithm. Then, accessibilities and attractiveness are computed according to Eqs. 3 and 4.
- **Iterative process (LUT for $N = 1, 3, 5, \dots, 27, 29$ million):** The algorithm performs the LUT model for the current population N , obtaining the output solutions. Given that the model holds memory of past buildings, the resulting supplies S_{vi} are used as inputs for the following iteration ($N = N + 2$). It is worth noting that the solution of the LUT model for each population level does not reach equilibrium, because the algorithm executes the LUT interaction twice for each population level.

Figure 3 summarizes how the algorithm proceeds, including running CUBE Land, MTE, and GDT models twice for each population N and the identification of the urban limit, which is denoted as Z^N .

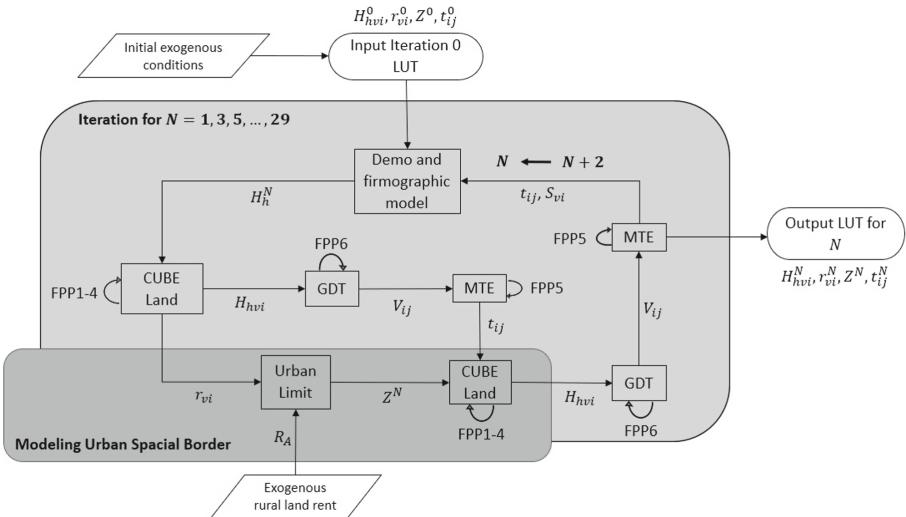


Fig. 3. General algorithm scheme

4 Simulation Results

A computational implementation was carried out to develop a highly simplified fictional scenario, called the base scenario, as a proof of concept of the model,

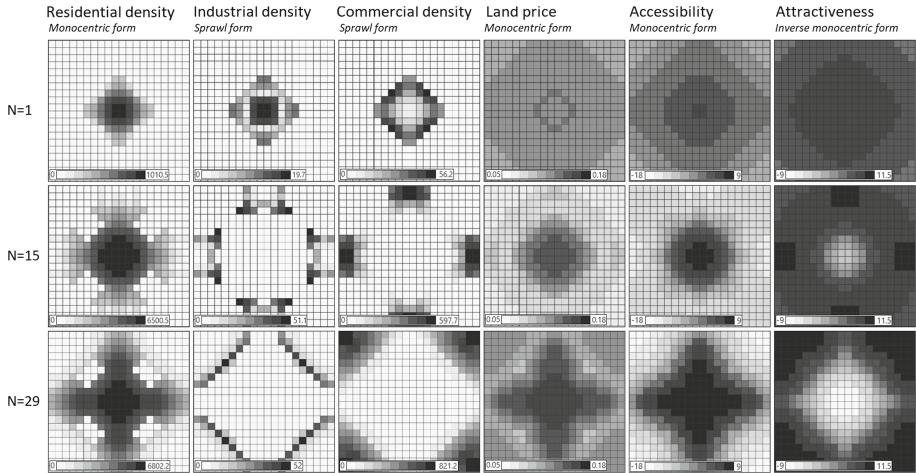


Fig. 4. Evolution of attributes for the original scenario

where the city evolves exclusively as a result of agents' behavior and market rules, using a linear bid function on densities (Eq. 6). Figure 4 shows the results obtained for the zonal variables in three population steps: 1, 15, and 29 million. We observe the following emerging urban forms:

- i. Residential density grows from an initially monocentric shape to a cross shape following the transport highways. It depicts four rural (white) areas in each quarter's center (as rural use is more profitable to the landowner).
- ii. Industrial and commercial land use follows an extreme tendency to locate at the city outskirts, as an outcome of the high value set for bids parameters of these activities associated with agglomeration and land size (Table 3).
- iii. Land prices are highly monocentric and cross-shaped around the highway, but at $N = 29$ million, the outskirt location of non-residential land use increases land prices of the corners of the grid, leaving lower rents in each quarter's center.
- iv. The spatial distribution of accessibility is monocentric combined with the cross shape of highways, while attractiveness is inverse monocentric.

A principal objective was to identify if the city evolves symmetrically despite the system complexity and the memory of the built area along with its population and size growth. As shown in Fig. 4, symmetry is apparent, but we report that the system is highly sensitive to very small asymmetrical values (of the order of 10^{-10}) in the MTE estimations of travel time, such that the observed symmetry is obtained by rounding the model's values of trips to represent meaningful values of trips (vehicles per hour). Without rounding, the initial small asymmetry grows with population inducing asymmetric location patterns. Of course, symmetry is not a realistic feature of real cities, but it is a necessary property of the model under the fictitious conditions set in this scenario.

Additionally, Fig. 5 depicts aggregate rents versus population in a logarithmic scale, showing that aggregate rents grow in a super-linear way with the population with a scale parameter of 1.09 approximately. This result demonstrates that the power law model $y = N^\beta$, supported by data [10] and explained by the theoretical microeconomic model [5] implemented in the VLT-LUT software, is also replicated by our simulation results. It remains to analyze the effect on the scale parameters λ and μ of logit probabilities, because they are theoretically related [5].

Then, two types of modifications were made to the base scenario: case (1), changes in the transport network; case (2), changes in the bid function on densities. In case (1) we increase the capacities of certain roads connected to the highways, while in case (2) we consider a superlinear function of densities followed by a saturation effect, i.e., after the saturation level, the density decreases its growth rate on bids functions.

First, regarding the attributes evolution, case (1) ends (a population of 29 million) with practically identical distributions as the base scenario. On the other hand, case (2) presents substantial differences, as the growth of the urban special border is significantly faster, such that every zone becomes urban by the population level of 19 million inhabitants (in the base scenario and case (1) there are remaining rural zones), while the allocation of agents is more homogeneous than those of the base scenario. Second, regarding the rent scale parameter, in both cases, it decreases: in case (1) to 1.07, close to the base scenario, while in case (2) to 1.013, a stronger decrease (Fig. 5).

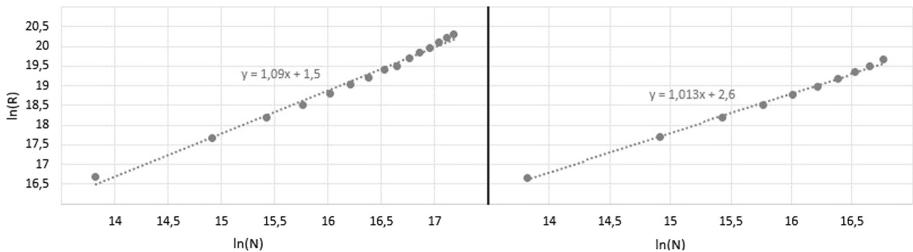


Fig. 5. Scaling of aggregate rents for base scenario and case (2)

5 Final Comments and Conclusions

We conclude that the set agglomeration economies and transportation network parameters may induce spatial segregation of residential and non-residential activities. While the modeled scenarios are somehow unrealistically extreme, it is useful to observe the impact of the different sensitivities to agglomeration economies and transportation costs.

A second important conclusion is that the transport network shapes the city form. In the performed simulations, we let the transport network be highly simplified (private and with only one cross of highways) to emphasize how transport

capacity induces location patterns. Thus, the urban form evolves from a small monocentric city to a shape that results from the combination of monocentric and transportation network shapes while agglomeration economies generate patterns of segregated agents. The emerging lesson is that roads infrastructure development is a policy that significantly shapes cities, while agglomeration economies are endogenous effects resulting from agents' behavior, which is exogenous to urban policies, although they may also be modified by subsidy policies.

It is important to highlight that this model is intrinsically dynamic, as it is not just a successive simulation of a city with different population levels, but a dynamically dependent process with memory (rural zones becoming residential, partial building demolition, and agglomeration patterns of specific agents).

Regarding future research, the plan is to simulate combined policies on zoning and road networks to obtain lessons about their long-term impacts and to analyze how the scaling law of rents is affected by them. A worthwhile extension of the software is to integrate, first, a more complex economic system with a labor market and, then, to integrate a system of cities in a region (as in [5]).

Acknowledgement. The authors gratefully acknowledge financial support from ISCI: grant ANID PIA/APOYO AFB180003.

References

1. Alonso, W.: *Location and Land Use*. Harvard University Press, Cambridge (1964)
2. Bailion, J.B., Cominetti, R.: Markovian traffic equilibrium. *Math. Program.* **111**(1), 33–56 (2008)
3. McGill, S.M.: Theoretical properties of biproportional matrix adjustments. *Environ. Plan. A* **9**, 687–701 (1977)
4. Martínez, F.J.: Cities' power laws: the stochastic scaling factor. *Environ. Plan. B Plan. Des.* **43**, 257–275 (2015)
5. Martínez, F.J.: *Microeconomic Modeling in Urban Science*. Academic Press, Cambridge (2018)
6. Pagliara, F., Preston, J., Simmonds, D.: *Residential Location Choice*. Springer, Heidelberg (2010). <https://doi.org/10.1007/978-3-642-12788-5>
7. SECTRA-MIDEPLAN, R.: Construcción de planes estratégicos de desarrollo del S.T.U. del Gran Valparaíso, IV etapa. Biblioteca SECTRA (2003)
8. SECTRA-MIDEPLAN, R.: Análisis, desarrollo y evaluación de proyectos urbanos. III etapa. Efectos urbanos futuros en Santiago EFFUS. Biblioteca SECTRA (2008)
9. West, G.B.: The origin of universal scaling laws in biology. *Physica A* **263**(1), 101–113 (1999)
10. West, G.B.: *Scale: The Universal Laws of Life and Death in Organisms, Cities and Companies*. Hachette UK (2017)
11. West, G.B., Brown, J., Enquist, B.: The fourth dimension of life: fractal geometry and allometric scaling of organisms. *Science* **284**(5420), 1677–1679 (1999)
12. West, G.B., Brown, J., Enquist, B.: A general model for ontogenetic growth. *Nature* **413**, 628–31 (2001)



Layered Hodge Decomposition for Urban Transit Networks

Unchitta Kan^(✉) and Eduardo López

George Mason University, Fairfax, VA 22030, USA

ukanjana@gmu.edu

<https://unchitta.com>

Abstract. Modeling the amount of passenger flow along any given line segment of an urban transit network is a challenging task due to the complexity of the system. In this paper, we embark on a characterization of these flows on the basis of a combination of (1) a layered decomposition of the origin-destination matrix, and (2) the Hodge decomposition, a discrete algebraic topology technique that partitions flows into gradient, solenoidal, and harmonic components. We apply our method to data from the London Underground. We find that the layered decomposition estimates the contribution of each origin-destination pair to the flow on each network link, and that the solenoidal and harmonic flows can be described by simple equations, thereby reducing much of the solution to determining gradient flows. Our exploratory analysis suggests it may be feasible to develop solution methods for the transit flow problem with a complexity equivalent to the solution of a hydraulic or electric circuit.

Keywords: Urban transit networks · Network flows · Link flow analysis and predictions · Hodge decomposition · Discrete algebraic topology

1 Introduction and Background

The prediction of patterns of passenger flow inside transit networks can be a challenging task due to the complexity of the network topology, its rules of functioning (time-tables, fares, loading capacities), and passengers' route preferences [1]. The origin of most of the difficulty stems from the feedback that emerges when individuals simultaneously crowd the network with trips going between a multiplicity of origins and destinations (so-called multi-commodity flows), generating congestion and a subsequent set of individual decisions that further affect said congestion. Over a long history of research [2], a multitude of ever-developing approaches have been applied to address this challenge, including network techniques [3–6], the theory of fluid flow [7], principles of time minimization [8,9], optimization methods [10,11], spatial coarse-graining [3], and agent-based modelling [12], among others. A useful review of the state of the literature can be found in [13]. Although algorithms in the literature are very

powerful and well developed, limitations exist. Some algorithms generate spurious cyclic solutions [14], others capture certain behavior at the expense of others [15], and yet many results only address limit theorems. Fully analytical methods do not exist that can capture the realities of these transit systems.

In this paper, we develop an analytic decomposition of observed flows in a transit network based on a combination of a layered separation of flows and Hodge theory. First, each layer in our method is made up of the network structure together with a single source node from which flow is generated. By employing a route choice method, the source node distributes its flows all across the network to other destination nodes. Then, Hodge decomposition is used to produce a principled partition of the flow into gradient, solenoidal, and harmonic parts; each flow type travels the network in a distinct manner. We make the critical observation that both solenoidal and harmonic flows in each layer can be approximated well by simple analytic expressions that do not require complicated solution methods. This suggests that, starting from an origin-destination matrix (see below, also [1]) of passengers in a transit system, it is possible to solve only the set of gradient flows to obtain an approximation for the flow on each segment (link) of the network.

The process presented here is advantageous for studying the transit flow problem because (1) the data required (the origin-destination matrix and the network topology) to estimate the Hodge flows are generally easy to acquire; and (2) the solution method, requiring little more than a solution of a circuit problem and the use of analytical approximations, offers a robust and simple path to estimation of transit network flows. Although our results have been developed as a proof of principle and have a number of simplifying assumptions, they show consistency with the principles of network flows, and further development is warranted in order to create an easily applicable method of approximation of transit network flows.

Hodge decomposition has been applied in a variety of settings like statistical ranking [16], topological data analysis [17], and information and money flows [18, 19]. However, as far as the authors are aware, this method has not been applied to the study of transit flows. Given its ability to tackle quite general types of flow in a network, we believe this offers a unique opportunity to explore its power when deployed in the context of transit flows.

This paper is organized as follows. The rest of this section describes the detailed setting of our problem and the data we use to explore it. Then, in Sect. 2 we provide a formal description of Hodge decomposition, defining the key equations that need to be used, and describe our application of the method to a layered decomposition of a transit network. Finally, we describe the results and findings from applying our method to London Underground data, as well as provide a discussion and conclusion in Sect. 3.

1.1 Transit Systems Flow and the Route Assignment Problem

In its simplest representation, a transit system can be seen as a network G , with n nodes and m links. Although such networks have directed links, it is common for each pair of nodes i and j in the system to be navigable in both directions.

This fortunate bi-directionality is a necessary element of Hodge decomposition. Since this paper is concerned with the explanation of flows, our analysis can be done without reference to other complicating factors of transit systems such as fares, schedules, or vehicle capacities.

As individuals enter the system via some node i of G and travel to their destinations, they collectively generate traffic flows. Data for these flows are generally found in two forms: the least detailed form is given by origin-destination (OD) matrices \mathbf{O} , where \mathbf{O}_{ij} is equal to the number of trips that start at i and end at j over a specified period of time (i and j need not be directly connected by a link); the more specific form is given by values of *link* (or segment/edge) flows t_{ij} , i.e. the number of travelers along a specific link (i, j) of G over a specified time period, where the direction of travel is $i \rightarrow j$. Note that in general $\mathbf{O}_{ij} \neq \mathbf{O}_{ji}$ and $t_{ij} \neq t_{ji}$.

The fundamental connection between \mathbf{O} and the set of segment flows t_{ij} emerges from passengers' choices on how they travel from their origin to destination. Route assignment has generated a large body of research, but whose considerable advancements have led to increasing dependence on complicated computational models [13, 14]. However, OD matrices are much easier to obtain, especially with wide usage of electronic fare-collection cards like the Oyster card of Transport for London, whereas detailed segment traffic are often generated on the basis of very detailed models (often solutions of the so-called route choice or assignment problem) or surveys. The disparity in the ease with which these data can be obtained makes it imperative to maximize the amount of information that can be extracted from just OD matrices, which is our goal in this study.

1.2 Data

Because a great deal of data has been made publicly available for the London Underground (LU) system, and the network itself is relatively large and exhibits non-trivial structure containing cliques and loops, the system makes a good candidate for the empirical exploration of our method. To study the LU passenger flows, we rely on a combination of data sources.

First, the network topology and spatial data are obtained from <https://github.com/oobrien/vis>. These data rely on OpenStreetMap™ and contain a list of LU stations, their latitude and longitude coordinates, and a list of segments connecting the stations and their coordinates. We convert the cartographic data into a network representation of the LU system that also includes information about the actual travel length of each segment (as opposed to crow-fly distance).

Next, the Rolling Origin-Destination Survey (RODS) periodically published by Transport for London (TfL) (<http://crowding.data.tfl.gov.uk/>) provides OD matrices describing the flow from each of the LU stations to all other stations by time of day in a typical weekday. Moreover, RODS also provides data on segment loading by train line and by time of the day in a typical weekday, from which we aggregate by train line to produce link flow data. In this study, we focus on average daily flows and therefore, both the OD matrix and link flow data are aggregated to a full day. We use the 2017 RODS data.

2 Methodology

2.1 Hodge Decomposition

The Hodge decomposition is a discrete analogue of the Helmholtz decomposition which rests upon the fundamental theorem of vector calculus. Given a general flow vector on a discrete topology such as a network, we can represent the flow as a sum of gradient (curl-free), solenoidal (divergence-free), and harmonic flows (both curl- and divergence-free) [20]. To better illustrate the application of this decomposition to flow data, we proceed to give a general introduction. For a more rigorous treatment of the topic, see [20, 21]. An enjoyable presentation of the topic with an applied flavor can be found in [22].

Let $G(V, E)$ represent a network of interest (for example, the LU network), where V is the set of n nodes and E the set of m undirected links. Depending on the exact set of links present, G may also exhibit triangles, i.e., triplets of nodes h, j, k forming a clique (all pairs of nodes of the triplet are connected by a link). In addition, the Hodge decomposition method requires the notion of orientation of the nodes, links, triangles, etc., which provides a convention of what ordering the nodes must have. This orientation specifies a positive direction for flows. For example, let $[u, v]$ be an oriented link ($[\cdot]$ is used to indicate orientation) where some ordering $u < v$ holds. Flow traveling on this link along the direction $u \rightarrow v$ would be represented by a positive value. The ordering convention can be chosen arbitrarily, as long as it is consistent.

Hodge decomposition rests on operators that perform the equivalent functions of the continuous operators involved in Helmholtz's Theorem. In that context, one uses divergence (usually $\nabla \cdot$ or simply div), curl ($\nabla \times$ or curl), and the gradient operator (∇ or grad). A complete set of discrete equivalents to the continuous operators is available, which we now present.

First, the gradient operator is given by the $m \times n$ coboundary-1 matrix \mathbf{A} ,

$$\mathbf{A}_{[v_0, v_1], u_0} = \begin{cases} 1 & \text{if } v_1 = u_0 \\ -1 & \text{if } v_0 = u_0 \\ 0 & \text{otherwise,} \end{cases}$$

where u_0, v_0, v_1 are nodes and $[v_0, v_1]$ an oriented link of G . If each node i is assigned some node potential value, say, η_i , the effect of \mathbf{A} on the column vector $\boldsymbol{\eta}$ is to generate flows $\eta_i - \eta_j$ along a link $[i, j]$. These are gradient flows, which we record in the $[i, j]$ row of column vector ϕ_g .

Second, the transposed matrix \mathbf{A}^T is perhaps more familiar than \mathbf{A} in the networks literature. It is the (oriented) incidence matrix of G . It is not difficult to show that \mathbf{A}^T is equivalent to the $-\text{div}$ (negative divergence) operator.

Third, the curl operator \mathbf{B} is obtained in a similar way, and is the transpose of the oriented edge-face incidence matrix (or the boundary-2 matrix)

$$\mathbf{B}_{[v_0, v_1, v_2], [u_0, u_1]} = \begin{cases} 1 & \text{if } v_1 = u_0, v_2 = u_1 \quad \text{or} \quad v_0 = u_0, v_1 = u_1 \\ -1 & \text{if } v_0 = u_0, v_2 = u_1 \\ 0 & \text{otherwise.} \end{cases}$$

The dimension of this matrix is $f \times m$, where f is the number of triangles (or so-called 2-simplices) in the network. The curl effect can be seen from applying \mathbf{B} to a flow column vector, say ϕ , where each element is a flow along a link. The transpose matrix \mathbf{B}^T has the effect of taking a column vector of network triangle potentials ψ and generating a column vector of link flows ϕ_s .

Theorem 1 (Cohomology). *The matrices \mathbf{A}^T and \mathbf{B}^T satisfy the cohomology property [20]*

$$\mathbf{A}^T \mathbf{B}^T = \mathbf{0}. \quad (1)$$

Similarly, through the definition of transpose of a matrix, the homology property is satisfied, i.e.

$$\mathbf{B} \mathbf{A} = \mathbf{0}. \quad (2)$$

This theorem tells us, in a more familiar calculus language, that $\text{div curl} = 0$ (i.e., curl is divergence free), and that $\text{curl grad} = 0$ (i.e., gradient is curl free).

To not lose sight of where we are headed, we must recall that the Helmholtz Theorem states that a vector field can be decomposed into a curl free divergent part and a divergence free rotational part. In discrete domains, the same basic notion applies. To construct these notions in discrete domains (i.e., networks), we take advantage of the linear algebra of the description.

Thus, for Hodge decomposition, our overall goal is to find a way to write any general flow ϕ in an *organized* fashion. Let us define \mathcal{F} , the space of all possible flows ϕ on the links of G . How could we decompose any flows in \mathcal{F} into parts that are mutually exclusive, thus giving an unambiguous decomposition? If such a decomposition exists, \mathcal{F} can be written as

$$\mathcal{F} = \mathcal{F}^{(1)} \oplus \cdots \oplus \mathcal{F}^{(q)},$$

where it is assumed that \mathcal{F} is decomposed into q mutually exclusive subsets of flows. Thus, any flow ϕ can be constructed as a sum of elements of each of the subsets.

Consider the set of flows $\mathcal{F}^{(g)}$ that can be obtained from applying \mathbf{A} to a node potential $\boldsymbol{\eta}$. This set is formally equal to the image set of \mathbf{A} , and thus can be written as $\mathcal{F}^{(g)} = \text{Image}(\mathbf{A})$. Let us denote by $\mathcal{F}^{(g^\perp)}$ the complement set of $\mathcal{F}^{(g)}$ with respect to \mathcal{F} , which includes all flows that cannot be generated from the application of \mathbf{A} to a node potential. From this perspective, $\mathcal{F} = \mathcal{F}^{(g)} \oplus \mathcal{F}^{(g^\perp)}$.

Another set of flows mentioned above were those given by the application of \mathbf{B}^T to a triangle potential ψ . Let us label this set as $\mathcal{F}^{(s)}$, which is formally given by $\mathcal{F}^{(s)} = \text{Image}(\mathbf{B}^T)$. Once again, it is also possible to define the complement set of $\mathcal{F}^{(s)}$, $\mathcal{F}^{(s^\perp)}$, and note that $\mathcal{F} = \mathcal{F}^{(s)} \oplus \mathcal{F}^{(s^\perp)}$. The cohomology or homology relations shown before (Eqs. 1 and 2) also led to the realization that if $\phi \in \mathcal{F}^{(g)}$, then it is not in $\mathcal{F}^{(s)}$. Similarly, if $\phi \in \mathcal{F}^{(s)}$, then it is not in $\mathcal{F}^{(g)}$. These relations imply that $\mathcal{F}^{(g)} \cap \mathcal{F}^{(s)} = \emptyset$. Therefore, \mathcal{F} can be decomposed as

$$\mathcal{F} = \mathcal{F}^{(g)} \oplus \mathcal{F}^{(s)} \oplus \mathcal{F}^{(gs^\perp)}.$$

This implies that a general flow ϕ can be written as a sum

$$\phi = \phi_g + \phi_s + \phi_h, \quad (3)$$

where ϕ_g is a purely gradient flow, ϕ_s is purely *solenoidal*, and ϕ_h is called *harmonic* flow. Harmonic flows are divergence-free flows that circulate the network beyond 2-simplices (hence they are also curl-free). Written in another way,

$$\phi = \mathbf{A}\eta + \mathbf{B}^T\psi + \phi_h, \quad (4)$$

where η_i and ψ_τ are called the “Hodge potential” of node i and triangle $\tau = [h, j, k]$, respectively.

To solve for the potentials of the nodes, note that

$$\begin{aligned} -\mathbf{A}^T\phi &= -\mathbf{A}^T\phi_g - \mathbf{A}^T(\phi_s + \phi_h) \\ &= -\mathbf{A}^T\phi_g = -\mathbf{A}^T\mathbf{A}\eta = -\mathcal{L}\eta, \end{aligned}$$

where \mathcal{L} is the Laplacian matrix of G . The matrix \mathcal{L} is singular (all its rows/columns add to 0), so only up to $n - 1$ potentials can be uniquely determined. Arbitrarily assuming that one of the nodes has $\eta = 0$ eliminates the corresponding row and column in the problem and generates a reduced matrix \mathcal{L}' that can be used to obtain the remaining η . Specifically, we solve

$$\eta = -\mathcal{L}'^{-1}(\operatorname{div} \phi), \quad (5)$$

where it is understood that only the unknown potentials and the respective flows that contribute to them are included. This is in essence the gradient problem solved by the Poisson equation ($-\mathbf{A}^T\phi$ provides the flow sources and sinks).

In a similar fashion, the triangle potentials can be determined from

$$\psi = (\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{B}\phi \quad (6)$$

which provides a solution based on the measured curl values of the flow, i.e., $\mathbf{B}\phi$. Finally, harmonic flows satisfy the homogeneous generalized Laplace equation

$$(\mathbf{A}\mathbf{A}^T + \mathbf{B}^T\mathbf{B})\phi_h = \mathbf{0}. \quad (7)$$

2.2 Layered Hodge Decomposition and Flow Prediction

Our goal is to describe and reconstruct link (segment) flows in the London Underground system, or any other transit system, from just the knowledge of the network structure and an OD matrix. Note that the Hodge decomposition as described above could not achieve this since the input it requires are all the measured flows on the network. In other words, from the stated inputs, Hodge decomposition can describe the flows but not predict them. To overcome this limitation, we explore an adapted version of the Hodge decomposition which we refer to as the *layered Hodge decomposition* (LHD) method.

In the LHD method, the problem is broken up into n layers, one for each network node. In layer L_i , there is only one source node, i , and the remaining nodes can only be sinks. The i th row sum $\|\mathbf{O}_i\|$ of the OD matrix tells us how many units of flow enter the system through i , and \mathbf{O}_{ij} tells us how many of those are absorbed by sink j . This layer approach allows us to use \mathbf{O} in a way that addresses two important effects: (1) the contribution of flow along any link due to travellers departing a particular node i , and (2) directionality of flows. For each layer, we can apply a method of choice to route the flow from an origin i to all other destinations. Since in this paper we are addressing a proof of concept, we apply the “all-or-nothing” solution method of shortest paths (weighted by segment lengths), but this choice is not necessary for the validity of our results.

To explain the intuition behind our approach, let us represent the link flows that result from solving the route assignment problem as $\varphi^{(L_i)}$, where $\varphi_{uv}^{(L_i)}$ is the flow on the oriented basis link $[u, v]$. The ability of our layered approach to address the contribution by a given origin can be seen intuitively: any non-zero flow $\varphi_{uv}^{(L_i)}$ will increase monotonically with $\|\mathbf{O}_i\|$. To see why the approach addresses the directionality problem, consider that if the solution method leads to $|\varphi_{uv}^{(L_i)}| > 0$, then formally $\varphi_{vu}^{(L_i)} = 0$, i.e., a link in the solution of a given layer is not traversed in both directions, $u \rightarrow v$ and $v \rightarrow u$. If this were the case, the method would not be optimizing flows. Violation of this assumption may be allowed in some methods with traveller uncertainty or error but this would only occur in rare cases. Thus, we neglect this effect.

Proceeding with our approach, $\varphi^{(L_i)}$ can be decomposed into the gradient, solenoidal, and harmonic flows of layer L_i . Critically, the application of Hodge decomposition to $\varphi^{(L_i)}$ gives us a set parameters (the Hodge potentials $\eta^{(L_i)}$, $\psi^{(L_i)}$, and the harmonic flow $\tilde{\phi}_h^{(L_i)}$) in each layer. We record these parameters and study their relationships with the network structure G and \mathbf{O} (or \mathbf{O}_i for layer L_i). If these relationships can be found, the entire flow $\varphi^{(L_i)}$ would be characterized only from G and \mathbf{O} . Symbolically, suppose that

$$\tilde{\eta}^{(L_i)} = \mathbf{f}(G, \mathbf{O}_i), \quad \tilde{\psi}^{(L_i)} = \mathbf{p}(G, \mathbf{O}_i), \quad \text{and} \quad \tilde{\phi}_h^{(L_i)} = \mathbf{e}(G, \mathbf{O}_i) \quad (8)$$

are our global approximations, valid in all of the layers of the Hodge potentials and the harmonic flows where $\mathbf{f}(\cdot)$, $\mathbf{p}(\cdot)$, and $\mathbf{e}(\cdot)$ are some functions of only the network structure and the OD matrix. Then, following Eq. 4 in the L_i layer, we can estimate the link flows as

$$\begin{aligned} \phi^{(L_i)} &\approx \mathbf{A}\tilde{\eta}^{(L_i)} + \mathbf{B}^T\tilde{\psi}^{(L_i)} + \tilde{\phi}_h^{(L_i)} \\ &= \mathbf{Af}(G, \mathbf{O}_i) + \mathbf{B}^T\mathbf{p}(G, \mathbf{O}_i) + \mathbf{e}(G, \mathbf{O}_i), \end{aligned} \quad (9)$$

where again we assume that each link (u, v) should be traversed only in one direction and that should be reflected by the sign of $\phi_{uv}^{(L_i)}$. Finally, we can provide estimates \tilde{t}_{uv} and \tilde{t}_{vu} of directed link flows in the system at study by adding up our estimations across all of the layers, or

$$\begin{aligned} \tilde{t}_{uv} &= \sum_i \phi_{uv}^{(L_i)} \times \theta(\phi_{uv}^{(L_i)}), \\ \tilde{t}_{vu} &= \sum_i -\phi_{uv}^{(L_i)} \times \theta(-\phi_{uv}^{(L_i)}), \end{aligned} \quad (10)$$

where $\theta(\cdot)$ is the Heaviside step function.

Equation 10 provides a road map of our method. We need to find \mathbf{f} , \mathbf{p} , and \mathbf{e} . The first of these, \mathbf{f} , is actually quite straightforward and formally a function of \mathbf{O} and G . That is because \mathbf{f} is the solution to the classic gradient problem of hydraulic or electric circuits, where $-\boldsymbol{\eta}$ corresponds to hydrostatic pressure or electric potential. The challenge left is to determine the contributions from the last two terms and, as we show below, this can be estimated from knowledge of \mathbf{O} and G .

We must clarify that commuters do not travel literally as calculated from either the gradient, solenoidal, or harmonic flows. These flows are merely mathematical components that, when added, reproduce a desired overall flow. Commuters typically do their best to reduce their travel time, generally favoring straight routes. Such routes do not look like each of the individual components of Hodge flows, which have other distributions in a network: gradient flows expand through all the available network in, sometimes, long and inefficient paths, while solenoidal and harmonic flows literally circulate inside, respectively, triangles or longer loops. It is only the sum of the flows that is able to approximate the actual commuter behavior.

It is important to remind the reader that the estimated flows \tilde{t}_{uv} depend on the specific route choice algorithm used to solve each layer. Since we are attempting to illustrate a proof of concept, it is reasonable to start from a simple shortest path algorithm, which approaches the desired rider behavior. A more detailed analysis may suggest a better choice, but the principles of our findings should be robust to this change.

3 Exploratory Analysis Using London Underground Data

We now apply LHD to describe the flows in the LU using the network structure and OD matrix of trips between LU stations in a typical day in 2017 (see Sect. 1.2 for more details about the data). Distance to the King's Cross St. Pancras station is used as our convention for node ordering to determine link orientation.

To develop some intuition about the nature of the solution of the route choice algorithm used here, we first study the relation between t_{uv} measured for the LU, and $\sum_i \varphi_{uv}^{(L_i)}$ obtained from Dijkstra's shortest path algorithm (with segment lengths as weights) on each layer. An OLS regression with a zero y-intercept gives regression coefficient of 0.75 (s.e. 0.02) with $R^2 = 0.70$. This provides a basic idea of the representative power of this algorithm with respect to the actual flow data. However, since our immediate goal is not to strictly reproduce the flow for the LU, we do not dwell on this point any further.

To obtain the node potentials $\tilde{\boldsymbol{\eta}}^{(L_i)}$, the only inputs needed are the i th row of \mathbf{O} and G , which are then used in Eq. 5 to recover the desired solution. Note that any algorithm used to create flows along links layer by layer (the route choice algorithm) still needs to satisfy the same \mathbf{O} matrix, and therefore, $\tilde{\boldsymbol{\eta}}^{(L_i)}$ for all the layers is in fact *independent of the route choice algorithm!*

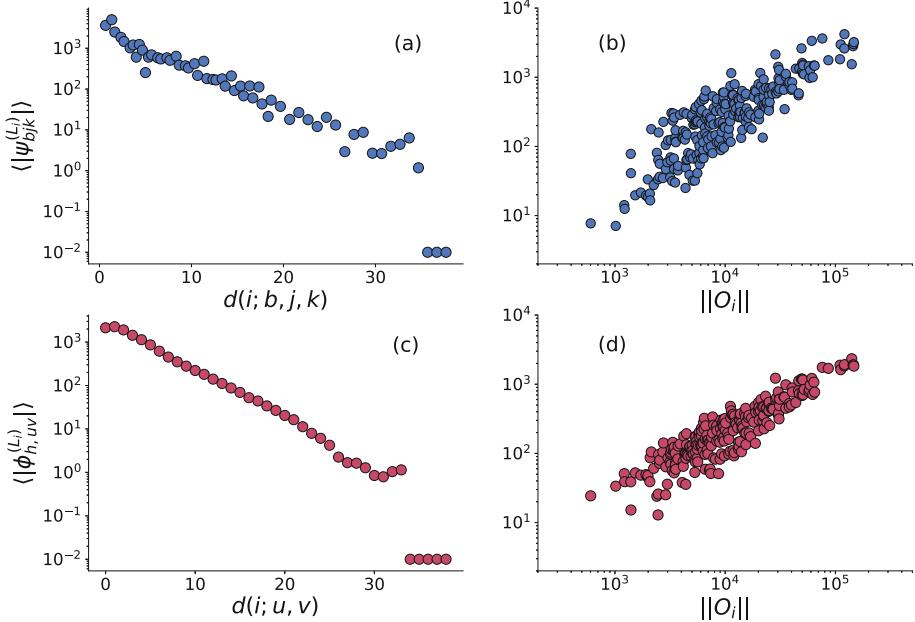


Fig. 1. Panels (a) and (c): Expectations of the solenoidal potential $|\psi_{bjk}^{(L_i)}|$ and harmonic flow $|\phi_{h,uv}^{(L_i)}|$. The first is conditioned on the average hop-count from the source node i to the nodes in the triangle (b, j, k) , and the second, from the source node i to link (u, v) , as measured by the hop-count from i to either u or v , whichever is closest. Panels (b) and (d): conditional expectations of $|\psi_{bjk}^{(L_i)}|$ and $|\phi_{h,uv}^{(L_i)}|$ given the magnitude of flow originating from the source node.

Going beyond the gradient flow, our analysis of the set of solenoidal potentials ψ in each layer leads to an approximate functional representation of ψ that can be inserted into Eq. 9. Concretely, we find that the magnitude of triangle potential b_{jk} in each layer, $|\psi_{bjk}^{(L_i)}|$, is a function of both the (hop-count) distance d from the source node to that triangle and the magnitude of input flow, $\|\mathbf{O}_i\|$, from the source node in that layer. The relationship with d is well-described by a log-linear function (Fig. 1a); with $\|\mathbf{O}_i\|$, the relation is linear (Fig. 1b). An equivalent functional form applies for harmonic flows $\tilde{\phi}_{h,uv}^{(L_i)}$ generated at i and passing through link $[u, v]$, with linear dependence on $\|\mathbf{O}_i\|$ and an exponential decay due to distance between the source node and link $[u, v]$. These empirical regularities allow us to approximate functional representations for the magnitudes $|\psi|$ and $|\phi_h|$, given by

$$|\tilde{\psi}_{bjk}^{(L_i)}| \approx |\mathbf{p}(G, \mathbf{O}_i)| = \kappa_1 \|\mathbf{O}_i\| \exp[-\kappa_2 d(i; b, j, k)], \quad (11)$$

$$|\tilde{\phi}_{h,uv}^{(L_i)}| \approx |\mathbf{e}(G, \mathbf{O}_i)| = \gamma_1 \|\mathbf{O}_i\| \exp[-\gamma_2 d(i; u, v)], \quad (12)$$

(Oxford Circus, Bond Street, Green Park) (Earls Court, West Kensington, Barons Court)

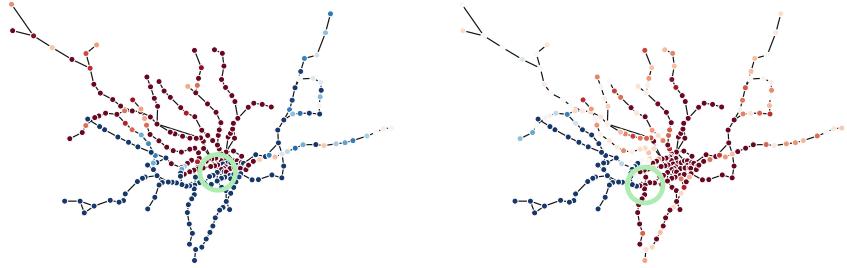


Fig. 2. A depiction of the value of the solenoidal potential ψ on two of the triangles in the LU network. When a node is colored red (blue), this means that in the layer where it is a source, the solenoidal potential on the triangle labeled in the title obtained from solving the LHD problem in that layer is negative (positive). When the color is grey, the potential value is 0. The approximate locations of the triangles are indicated via the green circles. Similar behaviors are observed for the rest of the triangles.

where the numerical constants are given by $\kappa_1 \approx 0.1$, $\kappa_2 \approx 0.1$, $\gamma_1 \approx 0.07$, and $\gamma_2 \approx 0.1$. These constants should depend on the route choice algorithm and other measures of G .

Note that both ψ_{bjk} or $\phi_{h,uv}$ can be positive, negative, or even 0. One approach to handle this is to tabulate the sign of each Hodge potential on each triangle and of harmonic flow on each link given layer L_i and source strength $\|\mathbf{O}_i\|$. These signs stay consistent if the time period underlying the OD matrix stays the same. However, one can also attempt to find a functional approximation of these signs. Here, we focus on $\psi_{bjk}^{(L_i)}$. Suppose $[b, j, k]$ has the orientation $b \rightarrow j, j \rightarrow k, k \rightarrow b$, and that the node which is closest to the source node i is b . Intuition and Fig. 2 suggest that if the flow out of i passes the triangle at all on the way to the sinks, we may expect $\psi_{bjk}^{(L_i)}$ to be negative if more flow travels in the $b \rightarrow k$ direction than in the $b \rightarrow j$ direction, because the former is against the orientation of $[b, j, k]$. Based on this reasoning, we develop an estimation of $\text{sign}(\psi_{bjk}^{(L_i)})$ using only topological and OD matrix information:

$$\text{sign}(\psi_{bjk}^{(L_i)}) = \begin{cases} -1 & \text{if } \underset{\{b,j,k\}}{\text{argmin}} \delta(i, x) = b \text{ and } c_{iV}(b, k) > c_{iV}(b, j), \\ -1 & \text{if } \underset{\{b,j,k\}}{\text{argmin}} \delta(i, x) = j \text{ and } c_{iV}(b, j) > c_{iV}(j, k), \\ -1 & \text{if } \underset{\{b,j,k\}}{\text{argmin}} \delta(i, x) = k \text{ and } c_{iV}(j, k) > c_{iV}(b, k), \\ 0 & \text{if } c_{iV}(b, j) = c_{iV}(j, k) = c_{iV}(b, k) = 0, \\ +1 & \text{otherwise,} \end{cases} \quad (13)$$

where $\delta(i, x)$ is the path length from node i to x using segment lengths, and

$$c_{iV}(e) = \sum_{v \in V} \frac{\sigma(i, v | e)}{\sigma(i, v)} \mathbf{O}_{iv}, \quad (14)$$

where V is the set of nodes in the network, $\sigma(i, v)$ is the number of shortest (i, v) -paths, and $\sigma(i, v|e)$ is the number of those paths that pass through link e . Equation 14 is a weighted (“trip-loaded”) version of the edge betweenness centrality in [23]. Note that the numbers of paths $\sigma(i, v|e)$ and $\sigma(i, v)$ need to be consistent with the route choice algorithm: if one uses something other than shortest paths, then this new algorithm dictates the values of $\sigma(i, v|e)$ and $\sigma(i, v)$.

We achieve a 91% percent overall accuracy in predicting $\text{sign}(\psi_{bjk}^{(L_i)})$ using this estimation, with some triangles having as high as 100% accuracy across all the layers. The same intuition would dictate the determination the signs for the harmonic flows.

As a final step to determine each \tilde{t}_{ij} , we sum all the flows and potentials required in Eq. 9 and then apply those into Eq. 10.

3.1 Discussion and Conclusion

Our work suggests that the non-trivial flows observed in a transit network such as the London Underground do not generally require a full solution of the Hodge decomposition to be predicted. Instead, Eqs. 11 through 14 provide a way to approximate the solenoidal and harmonic flows using only the network structure and the OD matrix, both of which are trivially necessary for any solution of the problem. These flows can then be plugged into Eqs. 9 and 10 to approximate the observed network flows. This procedure avoids altogether the use of more demanding and less tractable algorithms.

In order to move the Hodge decomposition description of network flows into a method, we would require a systematic study of the variations of $\tilde{\psi}$ and $\tilde{\phi}_h$ at short time intervals in order to use this quantity to be able to handle faster flow adjustments happening, e.g., within the time frame of a day. As immediate future work, our analysis should also be tested against other methods as well as using other transit systems for a more thorough robustness check. However, we are optimistic because many urban transit networks, especially in monocentric cities, are structurally similar to, if not simpler than, the LU network (they tend to follow the hub-and-spoke model). On the whole, our exploratory analysis here should provide a convincing case for the viability of the novel method we have described and more efforts in this direction are thus warranted.

References

1. Ortúzar, J. de D., Willumsen, L.G.: *Modelling Transport*. Wiley, New Jersey (2011)
2. Krylatov, A., Zakharov, V., Tuovinen, T.: *Optimization Models and Methods for Equilibrium Traffic Assignment*. Springer, Cham (2020). <https://doi.org/10.1007/978-3-030-34102-2>
3. Barthélémy, M.: *The Structure and Dynamics of Cities: Urban Data Analysis and Theoretical Modeling*. Cambridge (2016)
4. Louail, T., et al.: Uncovering the spatial structure of mobility networks. *Nat. Commun.* **6**, 6007 (2015)

5. Cajueiro, D.O.: Optimal navigation in complex networks. *Phys. Rev. E* **79**, 046103 (2009)
6. Gallotti, R., Porter, M.A., Barthélémy, M.: Lost in transportation: information measures and cognitive limits in multilayer navigation. *Sci. Adv.* **2**(2), e1500445 (2016)
7. Lighthill, M.J., Whitham F.R.S.: On kinetic waves II. A theory of traffic flow on crowded roads. *Proc. Royal Soc. Ser A.* **229**(1178), 317–345 (1955)
8. Wardrop, J.G.: Some theoretical aspects of road traffic research. *Proc. Ins. Civil Eng.* **2**, 325–378 (1952)
9. Beckmann, M.J., McGuire, C.B., Winsten, C.B.: *Studies in the Economics of Transportation*. Yale University Press, New Haven (1956)
10. Spiess, H., Florian, M.: Optimal strategies: a new assignment model for transit networks. *Transp. Res. B* **23B**(2), 83–102 (1989)
11. De Cea, J., Fernández, J.E.: Transit assignment to minimal routes: an efficient new algorithm. *Traffic Eng. Control* **30**, 491–494 (1989)
12. Legara, E.F., Monterola, C., Lee, K.K., Hung, G.G.: Critical capacity, travel time delays and travel time distribution of rapid mass transit systems. *Phys. A* **406**, 100–106 (2014)
13. Liu, Y., Bunker, J., Ferreira, L.: Transit: users' route-choice modelling in transit assignment: a review. *Transp. Rev.* **30**(6), 753–769 (2010)
14. Janson, B.N., Zozaya-Gorostiza, C.: The problem of cyclic flows in traffic assignment. *Transp. Res. B* **21B**(4), 299–310 (1987)
15. Raveau, S., Muñoz, J.C., de Grange, L.: A topological route choice model for metro. *Transp. Res. A* **45**(2), 138–147 (2011)
16. Jiang, X., Lim, L.-H., Yao, Y., Ye, Y.: Statistical ranking and combinatorial Hodge theory. *Math. Program.* **127**(1), 203–244 (2011)
17. de Silva, V., Morozov, D., Vejdemo-Johansson, M.: Persistent Cohomology and Circular Coordinates. *Disc. Comput. Geom.* **45**, 737–759 (2011)
18. Haruna, T., Fujiki, Y.: Hodge decomposition of information flow on small-world networks. *Front. Neural Circuit.* **10**, 77 (2016)
19. Kichikawa, Y., Iyetomi, H., Iino, T., Inoue, H.: Community structure based on circular flow in a large-scale transaction network. *Appl. Netw. Sci.* **4**(1), 1–23 (2019). <https://doi.org/10.1007/s41109-019-0202-8>
20. Grady, L.J., Polimeni, J.R.: *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. Springer, Cham (2010)
21. Lim, L.-H.: Hodge Laplacians on Graphs. *SIAM Rev.* **62**(3), 685–715 (2020)
22. Johnson, J.L., Goldring, T.: Discrete Hodge theory on graphs: a tutorial. *Comput. Sci. Eng.* **15**(5), 42–55 (2013)
23. Brandes, U.T.: On variants of shortest-path betweenness centrality and their generic computation. *Soc. Networks* **30**(2), 136–145 (2008)



Correction to: Complex Networks & Their Applications X

Rosa Maria Benito, Chantal Cherifi, Hocine Cherifi, Esteban Moro,
Luis M. Rocha, and Marta Sales-Pardo

Correction to:

R. M. Benito et al. (eds.):

Complex Networks & Their Applications X, SCI 1073,

<https://doi.org/10.1007/978-3-030-93413-2>

In the original version of the book, the following belated correction has been incorporated: The volume number has been changed from 1016 to 1073 in the Frontmatter, Backmatter and Chapter opening pages. The book and the chapter have been updated with the change.

The updated original version of the book can be found at
<https://doi.org/10.1007/978-3-030-93413-2>

Author Index

A

- Abou Rida, Amani, 563
Aceituno, Pau Vilimelis, 390
Agarwal, Nitin, 166
Aida, Masaki, 54
Akbas, Esra, 277
Akin, Myles, 629
Aktas, Mehmet Emin, 277
Amhaz, Rabih, 563
Anamalamudi, Satish, 253
Anderson, Paul, 664
Aridhi, Sabeur, 549
Aroke, Hilary, 716
Attux, Romis, 688

B

- Bagdasar, Ovidiu, 452
Bagnoli, Franco, 124
Bahushruth, C. S., 3, 78
Barbhuiya, Ferdous Ahmed, 3, 15, 78
Barkoczi, Daniel, 28
Ben-Eliezer, Omri, 587
Benito, Rosa M., 757
Benito, Rosa María, 66
Biazzo, Indaco, 780
Borondo, Javier, 757
Boström, Henrik, 441
Boukhobza, Taha, 676
Bramson, Aaron, 769
Brandes, Ulrik, 207
Brede, Markus, 353
Buchanan, Ashley, 716

C

- Caballero, Rafael, 66
Carchiolo, Vincenza, 452
Carley, Kathleen M., 89
Castellano, Gabriela, 688
Castillo, Ariel, 792
Casu, Benedetto, 124
Cavallaro, Lucia, 452
Cebrian, Manuel, 28
Celestini, Alessandro, 315
Chaitanya, Meher, 207
Chazelle, Bernard, 474
Chertok, Andrey, 575
Chin, Peter, 587, 742
Citu, 641
Cohen, Sarel, 742
Colaiori, Francesca, 315
Cruickshank, Iain J., 89

D

- Dabke, Devavrat Vivek, 474
Dang, Trung V., 587
Darius, Philipp, 100
de Bonfioli Cavalcabo, Guido, 124
De Clerck, Bart, 132
de la Paz Guala, Ricardo, 792
De Meo, Pasquale, 452
Dey, Kuntal, 3, 15, 78
Donoso, Pedro, 792
Douglas, Michael R., 587

E

- Elliott, Fernanda M., 365
Enduri, Murali Krishna, 253

F

- Fei, Luo, 523
 Filho, Carlos A. Stefano, 688
 Finke, Jorge, 652
 Fionda, Valeria, 144
 Fiumara, Giacomo, 452
 Friedrich, Tobias, 742
 Fukushima, Shintaro, 428
 Fushimi, Takayasu, 536

G

- Gangemi, Aldo, 486
 Gao, Matthew, 365
 Gao, Yang, 365
 Grassia, Marco, 452
 Guarino, Stefano, 315
 Guazzini, Andrea, 124
 Guedj, Benjamin, 341
 Guo, Yixin, 629
 Gupta, Sameer, 3, 15, 78
 Gutiérrez, Leonel, 792

H

- Hajarathaiah, Koduru, 253
 Halim, Nafisa, 377
 Hancock, Matthew, 377
 Hashizume, Ayako, 54
 He, Mingxuan, 365
 Hershcovitch, Moshik, 742
 Hussain, Owais A., 217

I

- Islam, Md Kamrul, 549
 Ito, Shuta, 536

K

- Kan, Unchitta, 804
 Kanai, Ryoga, 428
 Katenka, Natallia, 716
 Kent, Trevor G., 301
 Khan, Arijit, 523
 Kirdemir, Baris, 166
 Kishore, Aparna, 241
 Kißig, Otto, 742
 Kitsak, Maksim, 265
 Kogut, Stephen, 716
 Kuhlman, Chris J., 241, 377
 Kuikka, Vesa, 112
 Kuzma, Richard, 89

L

- La, Richard J., 229
 Lauwens, Ben, 132
 Liotta, Antonio, 452
 Liu, Shu, 415

- López, Eduardo, 804
 Losada, Juan Carlos, 66

M

- Ma, Long, 265
 Machi, Dustin, 241
 Machi, Lucas, 241
 Mahtha, Sanjeet Kumar, 641
 Mai, Van Sy, 229
 Mangioni, Giuseppe, 452
 Marathe, Achla, 377
 Martínez, Francisco, 792
 Mastrostefano, Enrico, 315
 Mathioudakis, Michael, 511
 Matta, John, 703
 McCulloh, Ian, 301
 Mees, Wim, 132
 Merchant, Arpit, 511
 Migler, Theresa, 181, 664
 Milli, Letizia, 329
 Mongiovì, Misael, 486
 Monod, Anthea, 461
 Moreno, Guillermo Romero, 353
 Mori, Larissa, 496
 Mozumder, Pallab, 377
 Murgas, Kevin A., 616

N

- Nagar, Seema, 3, 15, 78
 Nasuto, Slawomir J., 688
 Nguyen, Viet, 664
 Nikolentzos, Giannis, 441
 Nikoloski, Zoran, 605
 Nishiguchi, Mao, 415

O

- Omranian, Sara, 605

P

- Pansanella, Valentina, 329
 Parrend, Pierre, 563
 Pasulka, Alexis, 664
 Patsolic, Heather G., 301
 Pavon-Harr, Viveca, 301
 Pescetelli, Niccolo, 28
 Petit, Julien, 132
 Petrov, Tatjana, 289
 Pham, Minh An Antti, 112
 Pham, Tuan, 42
 Phillips, Nolan E., 301
 Pio-Lopez, Léo, 731
 Pirrò, Giuseppe, 144
 Porvatov, Vadim, 575
 Prasad, Abhishek, 641
 Prieto-Castrillo, Francisco, 757
 Pujol, Toyya A., 496

Q

Qureshi, Khizar, 156

R

Ramírez, Óscar, 652
Ravi, S. S., 241, 377
Robles, José Manuel, 66
Rocha, Camilo, 652
Rocha, Luis E. C., 132
Rodrigues, Paula G., 688
Romero, Miguel, 652
Rossetti, Giulio, 329

S

Salii, Yaroslav V., 194
Sandhu, Romeil, 616
Sanjel, Prashant, 703
Sato, João R., 688
Saucan, Emil, 461, 616
Semenova, Natalia, 575
Shankar, Shiv, 401
Sigbeku, John, 461
Simkin, Michael, 587
Smail-Tabbone, Malika, 549, 676
Solorzano, Ryan, 181
Soriano, Diogo C., 688
Stephany, Fabian, 100
Suyama, Ricardo, 688

T

Takahata, André K., 688
Taraz, Martin, 742
Tognazzi, Stefano, 289

Toriumi, Fujio, 415
Towsley, Don, 401

U

Usui, Shohei, 415

V

Vaginay, Athénaïs, 676
Van Mieghem, Piet, 265
Van Utterbeeck, Filip, 132
Vazirgiannis, Michalis, 441
Vendeville, Antoine, 341
Ventresca, Mario, 496
Vullikanti, Anil, 377

W

Waddington, Daniel, 742
Wood, Andrew, 587, 742
Wu, Tianqi, 587
Wu, Tianxing, 523

X

Xu, Nancy, 441

Y

Yadav, Gitanjali, 641
Yamanishi, Kenji, 428
Yang, Qi, 156

Z

Zaidi, Faraz, 217
Zaman, Tauhid, 156
Zastrow, Lena Rebecca, 315
Zhou, Shi, 341