

LNCS 10716

Wil M. P. van der Aalst et al. (Eds.)

Analysis of Images, Social Networks and Texts

6th International Conference, AIST 2017
Moscow, Russia, July 27–29, 2017
Revised Selected Papers



Springer

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, Lancaster, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Friedemann Mattern

ETH Zurich, Zurich, Switzerland

John C. Mitchell

Stanford University, Stanford, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Dortmund, Germany

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max Planck Institute for Informatics, Saarbrücken, Germany

More information about this series at <http://www.springer.com/series/7409>

Wil M. P. van der Aalst · Dmitry I. Ignatov
Michael Khachay · Sergei O. Kuznetsov
Victor Lempitsky · Irina A. Lomazova
Natalia Loukachevitch · Amedeo Napoli
Alexander Panchenko · Panos M. Pardalos
Andrey V. Savchenko · Stanley Wasserman (Eds.)

Analysis of Images, Social Networks and Texts

6th International Conference, AIST 2017
Moscow, Russia, July 27–29, 2017
Revised Selected Papers



Springer

Editors

Wil M. P. van der Aalst 
Eindhoven University of Technology
Eindhoven, The Netherlands

Dmitry I. Ignatov 
National Research University Higher School
of Economics
Moscow, Russia

Michael Khachay 
Krasovsky Institute of Mathematics
and Mechanics
Ekaterinburg, Russia

Sergei O. Kuznetsov 
National Research University Higher School
of Economics
Moscow, Russia

Victor Lempitsky 
Skolkovo Institute of Science
and Technology
Moscow, Russia

Irina A. Lomazova 
National Research University Higher School
of Economics
Moscow, Russia

Natalia Loukachevitch 
Moscow State University
Moscow, Russia

Amedeo Napoli 
LORIA, Campus Scientifique
Vandœuvre lès Nancy, France

Alexander Panchenko 
University of Hamburg
Hamburg, Germany

Panos M. Pardalos 
University of Florida
Gainesville, FL, USA

Andrey V. Savchenko 
National Research University Higher School
of Economics
Nizhny Novgorod, Russia

Stanley Wasserman 
Indiana University
Bloomington, IN, USA

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Computer Science

ISBN 978-3-319-73012-7

ISBN 978-3-319-73013-4 (eBook)

<https://doi.org/10.1007/978-3-319-73013-4>

Library of Congress Control Number: 2017961808

LNCS Sublibrary: SL3 – Information Systems and Applications, incl. Internet/Web, and HCI

© Springer International Publishing AG 2018, corrected publication 2018

This work is subject to copyright. All rights are reserved 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, express 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.

Cover logo: The background of the conference logo on the cover was revised. A new conference logo has been added.

Printed on acid-free paper

This Springer imprint is published by Springer Nature

The registered company is Springer International Publishing AG

The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland



Preface

This volume contains the refereed proceedings of the 6th International Conference on Analysis of Images, Social Networks, and Texts (AIST 2017)¹. The previous conferences during 2012–2016 attracted a significant number of students, researchers, academics, and engineers working on interdisciplinary data analysis of images, texts, and social networks.

The broad scope of AIST made it an event where researchers from different domains, such as image and text processing, exploiting various data analysis techniques, can meet and exchange ideas. We strongly believe that this may lead to cross fertilisation of ideas between researchers relying on modern data analysis machinery. Therefore, AIST brought together all kinds of applications of data mining and machine learning techniques. The conference allowed specialists from different fields to meet each other, present their work, and discuss both theoretical and practical aspects of their data analysis problems. Another important aim of the conference was to stimulate scientists and people from industry to benefit from the knowledge exchange and identify possible grounds for fruitful collaboration.

The conference was held during July 27–29, 2017. The conference was organised in Moscow, the capital of Russia, on the campus of Moscow Polytechnic University². This year, the key topics of AIST were grouped into six tracks:

1. General topics of data analysis chaired by Sergei Kuznetsov (Higher School of Economics, Russia) and Amedeo Napoli (LORIA, France)
2. Natural language processing chaired by Natalia Loukachevitch (Lomonosov Moscow State University, Russia) and Alexander Panchenko (University of Hamburg, Germany)
3. Social network analysis chaired by Stanley Wasserman (Indiana University, USA)
4. Analysis of images and video chaired by Victor Lempitsky (Skolkovo Institute of Science and Technology, Russia) and Andrey Savchenko (Higher School of Economics, Russia)
5. Optimisation problems on graphs and network structures chaired by Panos Pardalos (University of Florida, USA) and Michael Khachay (IMM UB RAS and Ural Federal University, Russia)
6. Analysis of dynamic behaviour through event data chaired by Wil van der Aalst (Eindhoven University of Technology, The Netherlands) and Irina Lomazova (Higher School of Economics, Russia)

One of the novelties this year was the introduction of a new specialised track on process mining (Track 6).

¹ <http://aistconf.org/>.

² <http://mospolytech.ru/?eng>.

The Programme Committee and the reviewers of the conference included 167 well-known experts in data mining and machine learning, natural language processing, image processing, social network analysis, and related areas from leading institutions of 30 countries including Argentina, Australia, Austria, Belgium, Brazil, Canada, China, Croatia, Czech Republic, Denmark, Egypt, Finland, France, Germany, Greece, Hungary, India, Ireland, Japan, Lithuania, Norway, Portugal, Qatar, Romania, Russia, Spain, Ukraine, The Netherlands, UK, and USA. This year we received 127 submissions mostly from Russia but also from Algeria, Australia, Brazil, China, Finland, Germany, India, Iran, Kazakhstan, Latvia, Mexico, The Netherlands, Norway, Turkey, USA, and Vietnam.

Out of 127 submissions only 37 papers were accepted as regular oral papers. Thus, the acceptance rate of this volume was around 29%. In order to encourage young practitioners and researchers, we included 34 papers to the supplementary proceedings after their poster presentation at the conference. Each submission was reviewed by at least three reviewers, experts in their fields, in order to supply detailed and helpful comments.

The conference featured several invited talks and an industry session dedicated to current trends and challenges.

The keynote talk was presented by Andrzej Cichocki on “Bridge Between Tensor Networks and Deep Neural Networks: From Fundamentals to Real Applications.”

The invited talks were:

- Stanley Wasserman (Indiana University, USA), “Sensitivity Analysis of p^* and SAOM: The Effects of Missing Data on Parameter Estimates”
- Dirk Fahland (Eindhoven University of Technology, The Netherlands), “Process Mining: Past, Present, and Open Challenges”

Sergey Nikolenko from the St. Petersburg Department of the Steklov Mathematical Institute presented a tutorial on “Deep Learning for Natural Language Processing.”

The business speakers also covered a wide variety of topics³. We list those invited talks below:

- Iosif Itkin (Exactpro Systems), Industrial day opening. Keynote on “Exactpro and Data Analysis in London Stock Exchange: Capital Markets, Post Trade and Information Services”
- Dmitry Bugaychenko (OK.ru), “Odnoklassniki Data Science Research Initiative”
- Olga Megorskaya (Yandex), “Yandex Toloka: Crowdsource Your Data”
- Artur Kuzin (Avito), “AvitoNet: Computer Vision Service in Avito”
- Alexander Zhebrak, Arthur Kadurin, Daniil Polykovskiy (Insilico Medicine), “Artificial Intelligence for Drug Discovery”

We would like to thank the authors for submitting their papers and the members of the Programme Committee for their efforts in providing exhaustive reviews.

³ A detailed program of AIST 2017 Business Day can be found on a separate website: <http://aistconf.ru>.

We would also like to express our special gratitude to all the invited speakers and industry representatives.

We deeply thank all the partners and sponsors. Our golden sponsors is Exactpro. Exactpro, a fully owned subsidiary of the London Stock Exchange Group, specialises in quality assurance for exchanges, investment banks, brokers, and other financial sector organisations. Our special thanks goes to Springer for their help, starting from the first conference call to the final version of the proceedings. Last but not least, we are grateful to all the organisers, especially to Marina Danshina, and the volunteers, whose endless energy saved us at the most critical stages of the conference preparation.

Here, we would like to mention the Russian word “aist” is more than just a simple abbreviation (in Cyrillic) — it means a “stork.” Since it is a wonderful free bird, a symbol of happiness and peace, this stork gave us the inspiration to organise the AIST conference. So we believe that this young and rapidly growing conference will likewise be bringing inspiration to data scientists around the world!

October 2017

Wil van der Aalst

Dmitry Ignatov

Michael Khachay

Sergei Kuznetsov

Victor Lempitsky

Irina Lomazova

Natalia Loukachevitch

Amedeo Napoli

Alexander Panchenko

Panos Pardalos

Andrey Savchenko

Stanley Wasserman

Organisation

Programme Committee Chairs

Wil van der Aalst	Eindhoven University of Technology, The Netherlands
Michael Khachay	Krasovsky Institute of Mathematics and Mechanics of RAS, Russia and Ural Federal University, Ekaterinburg, Russia
Sergei Kuznetsov	National Research University Higher School of Economics, Moscow, Russia
Amedeo Napoli	LORIA CNRS, University of Lorraine, and Inria, Nancy, France
Victor Lempitsky	Skolkovo Institute of Science and Technology, Russia
Irina Lomazova	National Research University Higher School of Economics, Moscow, Russia
Natalia Loukachevitch	Computing Centre of Lomonosov Moscow State University, Russia
Alexander Panchenko	University of Hamburg, Germany and Université catholique de Louvain, Belgium
Panos Pardalos	University of Florida, USA
Andrey Savchenko	National Research University Higher School of Economics, Nizhny Novgorod, Russia
Stanley Wasserman	Indiana University, USA

Proceedings Chair

Dmitry Ignatov	National Research University Higher School of Economics, Moscow, Russia
----------------	--

Business Day Chair

Rostislav Yavorskiy	National Research University Higher School of Economics, Moscow, Russia
---------------------	--

Steering Committee

Dmitry Ignatov	National Research University Higher School of Economics, Moscow, Russia
Michael Khachay	Krasovsky Institute of Mathematics and Mechanics of RAS, Russia and Ural Federal University, Ekaterinburg, Russia

Alexander Panchenko
Rostislav Yavorskiy

University of Hamburg, Germany
National Research University Higher School
of Economics, Russia

Programme Committee

Mehwish Alam	Université Paris 13, France
Gabriela Arevalo	Universidad Austral, Argentina
Artem Babenko	Yandex, Russia
Jaume Baixeries	Universitat Politècnica de Catalunya, Spain
Artem Baklanov	International Institute for Applied Systems Analysis, Austria
Sergey Bartunov	National Research University Higher School of Economics, Moscow, Russia and DeepMind, UK
Timo Baumann	Universität Hamburg, Germany
Darina Benikova	University of Duisburg-Essen, Germany
Malay Bhattacharyya	Indian Institute of Engineering Science and Technology, India
Chris Biemann	University of Hamburg, Germany
Elena Bolshakova	Moscow State Lomonosov University, Russia and National Research University Higher School of Economics, Russia
Anastasia Bonch-Osmolovskaya	National Research University Higher School of Economics, Russia
Aurélien Bossard	Université Paris 8, France
Jean-Leon Bouraoui	CENTAL (Université Catholique de Louvain), Belgium
Joos Buijs	Eindhoven University of Technology, The Netherlands
Andrea Burattin	Technical University of Denmark, Denmark
Evgeny Burnaev	Institute for Information Transmission Problems of RAS, Russia
Aleksey Buzmakov	Inria, LORIA (CNRS, Université de Lorraine), Nancy, France and National Research University Higher School of Economics, Perm, Russia
Ignacio Cassol	Universidad Austral, Argentina
Artem Chernodub	Institute of Mathematical Machines and Systems of NASU, Ukraine
Vladimir Chernov	Institute for Image Processing of RAS, Russia
Ekaterina Chernyak	National Research University Higher School of Economics, Moscow, Russia
Marina Chicheva	Samara State Aerospace University, Russia
Bonaventura Coppola	Technische Universität Darmstadt, Germany
Hernani Costa	University of Malaga, Spain
Massimiliano de Leoni	Eindhoven University of Technology, The Netherlands
Boris Dobrov	Lomonosov Moscow State University, Russia

Sofia Dokuka	National Research University Higher School of Economics, Moscow, Russia
Florent Domenach	Akita International University, Japan
Alexey Drutsa	Lomonosov Moscow State University and Yandex, Russia
Mirela-Stefania Duma	University of Hamburg, Germany
Richard Eckart de Castilho	Technische Universität Darmstadt, Germany
Judith Eckle-Kohler	UKP Lab, Technische Universität Darmstadt, Germany
Maria Eskevich	Radboud University Nijmegen, The Netherlands
Dirk Fahland	Eindhoven University of Technology, The Netherlands
Stefano Faralli	University of Mannheim, Germany
Victor Fedoseev	Samara National Research University, Russia
Michael Figurnov	Skolkovo Institute of Science and Technology, Russia
Elena Filatova	City University of New York, USA
Kerstin Fischer	University of Southern Denmark, Denmark
Fedor Fomin	University of Bergen, Norway
Thomas Francois	Université catholique de Louvain, Belgium
Oleksandr Frei	Universitetet i Oslo, Norway
Edward K. Gimadi	Sobolev Institute of Mathematics of RAS, Russia
Ivan Gostev	National Research University Higher School of Economics, Moscow, Russia
Natalia Grabar	Université Lille 3 and CNRS, France
Dmitry Granovsky	Yandex, Russia
Alexey Gruzdev	Intel, Russia
Ivan Habernal	Technische Universität Darmstadt, Germany
Mena Habib	Maastricht University, The Netherlands
Marianne Huchard	Université Montpellier 2 and CNRS, France
Dmitry Ignatov	National Research University Higher School of Economics, Moscow, Russia
Dmitry Ilvovsky	National Research University Higher School of Economics, Moscow, Russia
Vladimir Ivanov	Innopolis University, Russia
Pei Jun	Hefei University of Technology, China
Anna Kalenkova	National Research University Higher School of Economics, Moscow, Russia
Nikolay Karpov	National Research University Higher School of Economics, Nizhniy Novgorod, Russia
Egor Kashkin	V. V. Vinogradov Russian Language Institute of RAS, Russia
Mehdi Kaytoue	LIRIS - INSA de Lyon, France
Alexander Kelmanov	Sobolev Institute of Mathematics of RAS, Russia
Oleg Khamisov	Melentiev Institute of Energy Systems of RAS, Russia
Andrey Kibzun	Moscow Aviation Institute, Russia
Edward Klyshinsky	HSE Moscow Institute of Electronics and Mathematics, Russia
Yury Kochetov	Sobolev Institute of Mathematics of RAS, Russia

Ekaterina Kochmar	University of Cambridge, UK
Sergei Koltcov	National Research University Higher School of Economics, St. Petersburg, Russia
Olessia Koltsova	National Research University Higher School of Economics, St. Petersburg, Russia
Jan Konecny	Palacky University, Czech Republic
Daniil Kononenko	Skolkovo Institute of Science and Technology, Russia
Natalia Konstantinova	University of Wolverhampton, UK
Andrey Kopylov	Tula State University, Russia
Mikhail Korobov	ScrapingHub Inc., Ireland
Anton Korshunov	Institute for System Programming of RAS, Russia
Evgeny Kotelnikov	Vyatka State University, Russia
Ilias Kotsireas	Wilfrid Laurier University, Canada
Olga Krasotkina	Lomonosov Moscow State University, Russia
Tomas Krilavicius	Vytautas Magnus University, Lithuania
Victor Kulikov	Institute of Automation and Electrometry of RAS, Russia
Valentina Kuskova	National Research University Higher School of Economics, Moscow, Russia
Andrey Kutuzov	Universitetet i Oslo, Norway
Andrey Kuzmin	Skolkovo Institute of Science and Technology, Russia
Andrey Kuznetsov	Samara State Aerospace University, Russia
Sergei Kuznetsov	National Research University Higher School of Economics, Moscow, Russia
Alexander Lazarev	Institute of Control Sciences of RAS, Russia
Florence Le Ber	Université de Strasbourg, France
Vadim Lebedev	Skolkovo Institute of Science and Technology, Russia
Victor Lempitsky	Skolkovo Institute of Science and Technology, Russia
Alexander Lepskiy	National Research University Higher School of Economics, Moscow, Russia
Benjamin Lind	Anglo-American School of St. Petersburg, Russia
Irina Lomazova	National Research University Higher School of Economics, Moscow, Russia
Natalia Loukachevitch	Lomonosov Moscow State University, Russia
Olga Lyashevskaya	National Research University Higher School of Economics, Moscow, Russia
Yury Malkov	Institute of Applied Physics of RAS, Russia
Luis Marujo	Carnegie Mellon University, USA, and Instituto Superior Técnico, Portugal
Sérgio Matos	Universidade de Aveiro, Portugal
Yelena Mejova	Qatar Computing Research Institute, Qatar
Nizar Messai	Université François Rabelais Tours, France
Tristan Miller	Technische Universität Darmstadt, Germany
Olga Mitrofanova	St. Petersburg State University, Russia
Evgeny Myasnikov	Samara National Research University, Russia
Sergey Nikolenko	Steklov Mathematical Institute, St. Petersburg, Russia

Vassilina Nikoulina	Xerox Research Center Europe, France
Damien Nouvel	Université Sorbonne, France
Dimitri Nowicki	Institute of Mathematical Machines and Systems of NASU, Ukraine
Panos Pardalos	University of Florida, USA
Georgios Petasis	National Centre for Scientific Research Demokritos, Greece
Stefan Pickl	Universität der Bundeswehr München, Germany
Lidia Pivovarova	University of Helsinki, Finland
Vladimir Pleshko	RCO, Russia
Hernan Ponce-De-Leon	fortiss GmbH, Germany
Alexander Porshnev	National Research University Higher School of Economics, Nizhniy Novgorod, Russia
Alexey Potapov	AIDEUS, Russia
Surya Prasath	University of Missouri-Columbia, USA
Uta Priss	Ostfalia Universiy of Applied Sciences, Germany
Oleg Prokopyev	University of Pittsburgh, USA
Artem Pyatkin	Novosibirsk State University and Sobolev Institute of Mathematics, Russia
Carlos Ramisch	Aix Marseille University, France
Alexandr Rassadin	KPMG, Russia and National Research University Higher School of Economics, Nizhniy Novgorod, Russia
Artem Revenko	Semantic Web Company GmbH, Austria
Evgeniy Riabenko	National Research University Higher School of Economics, Moscow, Russia
Martin Riedl	University of Hamburg, Germany
Alexey Romanov	University of Massachusetts Lowell, USA
Andrey Ronzhin	St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences, Russia
Alexandra Roshchina	Institute of Technology Tallaght, Ireland
Eugen Ruppert	Technische Universität Darmstadt, Germany
Christian Sacarea	Babes-Bolyai University, Hungary
Mohammed Abdel-Mgeed	Ain Shams University, Cairo
M. Salem	
Sheikh Muhammad Sarwar	University of Massachusetts Amherst, USA
Andrey Savchenko	National Research University Higher School of Economics, Nizhniy Novgorod, Russia
Friedhelm Schwenker	Ulm University, Germany
Alexander Semenov	National Research University Higher School of Economics, Moscow, Russia
Oleg Seredin	Tula State University, Russia
Andrey Shcherbakov	University of Melbourne, Australia
Oleg Slavin	Institute for Systems Analysis of Russian Academy of Sciences
Jan Snajder	University of Zagreb, Croatia

Henry Soldano	Université Paris 13, France
Tobias Staron	Universität Hamburg, Germany
Dmitry Stepanov	Program System Institute of RAS, Russia
Vadim Strijov	Computing Center of RAS, Russia
Maria Sukhareva	Goethe University Frankfurt, Germany
Diana Sungatullina	Skolkovo Institute of Science and Technology, Russia
Laszlo Szathmary	University of Debrecen, Hungary
Irina Temnikova	Qatar Computing Research Institute, Qatar
Diana Troanca	Babes-Bolyai University, Hungary
Christos Tryfonopoulos	University of the Peloponnese, Greece
Denis Turdakov	Institute for System Programming of RAS, Russia
Dmitry Ulyanov	Skolkovo Institute of Science and Technology, Russia
Dmitry Ustalov	Krasovskii Institute of Mathematics and Mechanics of RAS, Russia and Ural Federal University, Yekaterinburg, Russia
Evgeniya Ustinova	Skolkovo Institute of Science and Technology, Russia
Alexander Vakhitov	St. Petersburg State University, Russia
Wil van der Aalst	Eindhoven University of Technology, The Netherlands
Natalia Vassilieva	Hewlett Packard Enterprise, USA
Dmitry Vetrov	Moscow State University and National Research University Higher School of Economics, Moscow, Russia
Renato Vimieiro	Universidade Federal de Pernambuco
Ekaterina Vylomova	The University of Melbourne, Australia
Roman Yangarber	University of Helsinki, Finland
Rostislav Yavorsky	National Research University Higher School of Economics, Moscow, Russia
Marcos Zampieri	University of Wolverhampton, UK
Nikolai Zolotykh	University of Nizhniy Novgorod, Russia
Olga Zvereva	Ural Federal University, Russia

Additional Reviewers

Sujoy Chatterjee
 Anton Eremeev
 Aleksey Glebov
 Sergey Khamidullin
 Vladimir Khandeev

Sofya Kulikova
 Abhishek Kumar
 Alexander Plyasunov
 Vladimir Servakh

Organising Committee

Rostislav Yavorskiy (Conference Chair)	National Research University Higher School of Economics, Russia
Andrey Novikov (Head of Organization)	National Research University Higher School of Economics, Russia
Marina Danshina (Venue Organization and Management)	Moscow Polytechnic University, Russia
Anna Ukhanaeva (Information Partners and Communications)	National Research University Higher School of Economics, Russia
Anna Kalenkova (Visa Support and International Communications)	National Research University Higher School of Economics, Russia
Alexander Gnevshov (Venue Organization and Management)	Moscow Polytechnic University, Russia

Volunteers

Daniil Bannyh	Moscow Polytechnic University, Russia
Ksenia Belkova	Moscow Polytechnic University, Russia
Tatiana Mishina	Moscow Polytechnic University, Russia
Zahar Kuhtenkov	Moscow Polytechnic University, Russia
Maxim Pasynkov	Krasovsky Institute of Mathematics and Mechanics of RAS, Ekaterinburg, Russia
Ivan Poylov	Moscow Polytechnic University, Russia

Sponsors

Golden sponsor

Exactpro

Bronze sponsor

Springer

Keynote and Invited Talks

Bridge Between Tensor Networks and Deep Neural Networks: From Fundamentals to Real Applications

Andrzej Cichocki^{1,2}

¹ RIKEN Brain Science Institute, Japan

² Skolkovo Institute of Science and Technology, Russia
cia@brain.riken.jp

Abstract. Tensor decompositions (TD) and their generalizations tensor networks (TN) are promising, and emerging tools in Machine Learning (ML), especially in Deep Learning (DL), since input/output data outputs in hidden layers can be naturally represented and described as higher-order tensors and most operations can be performed using optimized linear/multilinear algebra.

I will present a brief overview of tensor decomposition and tensor networks architectures and associated learning algorithms. I will also discuss several applications of tensor networks in Signal Processing, Machine Learning, both in supervised and unsupervised learning and possibility of dramatic reduction of set of parameters in state-of-the arts deep CNN, typically, from hundreds millions to tens of thousands of parameters. We focus on novel (Quantized) Tensor Train-Tucker (QTT-Tucker) and Quantized Hierarchical Tucker (QHT) tensor network models for higher order tensors (tensors of order at least four or higher). Moreover, we present tensor sketching for efficient dimensionality reduction which avoid curse of dimensionality.

Tensor Train-Tucker and HT models will be naturally extended to MERA (Multiscale Entanglement Renormalization Ansatz) models, TTNS (Tree Tensor Network States) and PEPS/PEPO and other 2D/3D tensor networks, with improved expressive power of deep learning in convolutional neural networks (DCNN) and inspiration to generate novel architectures of deep and semi-shallow neural networks. Furthermore, we will be show how to apply tensor networks to higher order multiway, partially restricted Boltzmann Machine (RBM) with substantial reduction of set of learning parameters.

Keywords: Tensor networks · Deep learning · Tensor decompositions
Neural networks

References

1. Cichocki, A., Mandic, D.P., Lathauwer, L.D., Zhou, G., Zhao, Q., Caiafa, C.F., Phan, A.H.: Tensor decompositions for signal processing applications: from two-way to multiway component analysis. *IEEE Signal Process. Mag.* **32**(2), 145–163 (2015)

2. Cichocki, A., Lee, N., Oseledets, I.V., Phan, A.H., Zhao, Q., Mandic, D.P.: Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Found. Trends Mach. Learn.* **9**(4–5), 249–429 (2016)
3. Cichocki, A., Phan, A.H., Zhao, Q., Lee, N., Oseledets, I.V., Sugiyama, M., Mandic, D.P.: Tensor networks for dimensionality reduction and large-scale optimization: Part 2 applications and future perspectives. *Found. Trends Mach. Learn.* **9**(6), 431–673 (2017)

Process Mining: Past, Present, and Open Challenges

Dirk Fahland

Eindhoven University of Technology, The Netherlands
d.fahland@tue.nl

Abstract. Since the first algorithms for automatically discovering process models from event logs have been proposed in the late 1990ies the problem of obtaining insights into processes by mining from event logs gained growing attention. By now, the field has grown into a maturing discipline and industry has begun adopting process mining in regular operations, supported by several commercial process mining solutions that are available on the market.

In the early days of process mining, several algorithms for constructively discovering a process model from an event log were proposed, each algorithm pursuing unique principles for constructing a model. This first generation of process discovery techniques, which includes, for instance, the alpha-algorithm, paved the ground for process mining as a research discipline. As these algorithms were applied in practice, new research challenges showed up, sparking new results in both pre-processing event data and evaluating process models on event logs. In particular the latter deepened the understanding of the challenges in process mining and established a reliable feedback mechanism in process mining in the form of conformance checking. This feedback mechanism enabled researching the second generation of process mining techniques addressing a large variety of problems such as quality guarantees for discovered models, including the data perspective in discovered models, or discovering temporal logic constraints. In particular, the inductive miner family was seen as a new milestone as it provided a systematic way to develop process discovery algorithms with reliable results. Yet again, as these more capable techniques are being applied to the growing and more detailed event data recorded in practice, further unsolved challenges arise.

In the first part of my talk I will draw an arc from the early days of process mining to the current state of the art in process mining – highlighting central techniques and their impact on later developments. In the second part of my talk, I will then turn to what kinds of event data and challenges are being found in practice today, how existing process mining techniques fail to address them, and thus which open challenges and opportunities the process mining field offers also for researchers from other domains.

Keywords: Process mining · Information systems
Combined modeling paradigms · Event logs

Sensitivity Analysis of p^* and SAOM: The Effects of Missing Data on Parameter Estimates

Stanley Wasserman

Indiana University, USA

stanwass@indiana.edu

Abstract. Many studies use complicated network data sets. Take, for example, the Framingham Heart Study, which was never intended for use in network analyses, but whose family and friend contact information can be considered relational data and thus can be massaged into a social network. Such data sets are often sampled in various ways, but the effects of the inherent sampling design on the findings of the study are unknown. Nevertheless, the sampling design will certainly influence the results in some way. There is little published research on the impact of network sampling on network structures and structural measures. There is even less research that investigates how network sampling impacts models that link network structure with behaviors and attitudes. We need not only to study how sampling designs for social network studies impact network measures, but also investigate how these sampling designs impact models that include social influence effects. The old studies of measurement error in network analysis (circa 1975) are woefully old. Our research contributes much-needed, but rarely discussed, important information to a rapidly growing field. Specifically, we study the following methodological questions:

1. How does eliminating links bias cross-sectional and longitudinal parameter estimates and statistical models? Missing links approximate both fixed choice designs where only a portion of links are provided and situations where respondents do not report all relationships that exist.
2. How does eliminating alters and their links bias cross sectional and longitudinal parameter estimates and statistical models? Missing alters represent panel-type loss of entire groups from a study as well as intermittent joiners and leavers in studies (alters that appear at multiple, but not sequential time points, as if a student were enrolled in a study and appeared at baseline but then not again until the fourth and fifth waves of a five wave study).
3. How does the method of sampling bias estimates and models? That is, do missing edges lead to greater problems than missing nodes? Does churn in these models lead to greater problems than the loss of alters at baseline due to sampling issues? Churn here refers to the presence of intermittent joiners and leavers.
4. How do systematic sampling approaches such as snowball sampling or link-tracing bias estimates and models? These approaches approximate respondent-driven sampling approaches and other trace based designs.

5. How does the information that is lost when one has personal (egocentered), rather than complete, network information bias estimates and models?
6. Does missingness in alter attribute variables have the same effect as missingness in structural variables?
7. How much does measurement error (caused, for example, by forced fixed-choice designs) affect statistical findings from these new models, particularly for social influence parameters?

Keywords: Social networks · Exponential-family random graph models
Stochastic actor-oriented models · Sensitivity analysis · Missing data

Deep Learning for Natural Language Processing (Tutorial)

Sergey Nikolenko^{1,2,3}

¹ St.-Petersburg Department of the Steklov Mathematical Institute, Russia

² National Research University Higher School of Economics, Russia

³ Academic University, Russia

sergey@logic.pdmi.ras.ru

Abstract. Over the last decade, deep learning has revolutionized machine learning. Neural network architectures have become the method of choice for many different applications. In this tutorial, we survey the applications of deep learning to natural language processing (NLP) problems.

We begin by briefly reviewing the basic notions and major architectures of deep learning, including some recent advances that are especially important for NLP.

Then we survey distributed representations of words, showing both how word embeddings can be extended to sentences and paragraphs and how words can be broken down further in character-level models.

Finally, the main part of the tutorial deals with various deep architectures that have either arisen specifically for NLP tasks or have become a method of choice for them; the tasks include sentiment analysis, dependency parsing, machine translation, dialog and conversational models, question answering, and other applications.

Keywords: Deep learning · Natural language processing
Machine learning applications

Contents

Natural Language Processing

Automated Detection of Adverse Drug Reactions from Social Media Posts with Machine Learning	3
<i>Ilseyar Alimova and Elena Tutubalina</i>	
Automated Detection of Non-Relevant Posts on the Russian Imageboard “2ch”: Importance of the Choice of Word Representations	16
<i>Amir Bakarov and Olga Gureenkova</i>	
A Morphological Processor for Russian with Extended Functionality	22
<i>Elena I. Bolshakova and Alexander S. Sapin</i>	
SyntaxNet Errors from the Linguistic Point of View	34
<i>Oleg Durandin, Alexey Malafeev, and Nikolai Zolotykh</i>	
Size vs. Structure in Training Corpora for Word Embedding Models: Araneum Russicum Maximum and Russian National Corpus	47
<i>Andrey Kutuzov and Maria Kunilovskaya</i>	
Combining Thesaurus Knowledge and Probabilistic Topic Models	59
<i>Natalia Loukachevitch, Michael Nokel, and Kirill Ivanov</i>	
Russian-Language Question Classification: A New Typology and First Results	72
<i>Kirill Nikolaev and Alexey Malafeev</i>	
Domain Adaptation for Resume Classification Using Convolutional Neural Networks	82
<i>Luiza Sayfullina, Eric Malmi, Yiping Liao, and Alexander Jung</i>	
Fighting with the Sparsity of Synonymy Dictionaries for Automatic Synset Induction	94
<i>Dmitry Ustalov, Mikhail Chernosukov, Chris Biemann, and Alexander Panchenko</i>	
Men Are from Mars, Women Are from Venus: Evaluation and Modelling of Verbal Associations	106
<i>Ekaterina Vylomova, Andrei Shcherbakov, Yuriy Philippovich, and Galina Cherkasova</i>	

Rotations and Interpretability of Word Embeddings: The Case of the Russian Language	116
<i>Alexey Zobnin</i>	

General Topics of Data Analysis

HuGaDB: Human Gait Database for Activity Recognition from Wearable Inertial Sensor Networks	131
<i>Roman Chereshnev and Attila Kertész-Farkas</i>	

On Finding Maximum Cardinality Subset of Vectors with a Constraint on Normalized Squared Length of Vectors Sum	142
<i>Anton V. Eremeev, Alexander V. Kelmanov, Artem V. Pyatkin, and Igor A. Ziegler</i>	

Using Cluster Analysis for Characteristics Detection in Software Defect Reports	152
<i>Anna Gromova</i>	

A Machine Learning Approach to Enhanced Oil Recovery Prediction	164
<i>Fedor Krasnov, Nikolay Glavnov, and Alexander Sitnikov</i>	

An Approach to Establishing the Correspondence of Spatial Objects on Heterogeneous Maps Based on Methods of Computational Topology	172
<i>Sergey Eremeev, Kirill Kuptsov, and Semyon Romanov</i>	

Predicting Winning Team and Probabilistic Ratings in “Dota 2” and “Counter-Strike: Global Offensive” Video Games	183
<i>Ilya Makarov, Dmitry Savostyanov, Boris Lityakov, and Dmitry I. Ignatov</i>	

Bagging Prediction for Censored Data: Application for Theatre Demand	197
<i>Evgeniy M. Ozhegov and Alina Ozhegova</i>	

Original Loop-Closure Detection Algorithm for Monocular vSLAM	210
<i>Andrey Bokovoy and Konstantin Yakovlev</i>	

Analysis of Images and Video

Organizing Multimedia Data in Video Surveillance Systems Based on Face Verification with Convolutional Neural Networks	223
<i>Anastasiia D. Sokolova, Angelina S. Kharchevnikova, and Andrey V. Savchenko</i>	

Satellite Image Forgery Detection Based on Buildings Shadows Analysis.	231
<i>Andrey Kuznetsov and Vladislav Myasnikov</i>	

Nonlinear Dimensionality Reduction of Hyperspectral Data Using Spectral Correlation as a Similarity Measure	237
<i>Evgeny Myasnikov</i>	
Large-Scale Shape Retrieval with Sparse 3D Convolutional Neural Networks	245
<i>Alexandr Notchenko, Yermek Kapushev, and Evgeny Burnaev</i>	
Floor-Ladder Framework for Human Face Beautification	255
<i>Yulia Novskaya, Sun Ruoqi, Hengliang Zhu, and Lizhuang Ma</i>	
Array DBMS and Satellite Imagery: Towards Big Raster Data in the Cloud	267
<i>Ramon Antonio Rodriges Zalipynis, Evgeniy Pozdeev, and Anton Bryukhov</i>	
Impulsive Noise Removal from Color Images with Morphological Filtering	280
<i>Alexey Ruchay and Vitaly Kober</i>	
Optimization Problems on Graphs and Network Structures	
An Exact Polynomial Algorithm for the Outerplanar Facility Location Problem with Improved Time Complexity	295
<i>Edward Gimadi</i>	
Approximation Algorithms for the Maximum m -Peripatetic Salesman Problem	304
<i>Edward Kh. Gimadi and Oxana Yu. Tsidulko</i>	
A Randomized Algorithm for 2-Partition of a Sequence	313
<i>Alexander Kel'manov, Sergey Khamidullin, and Vladimir Khandeev</i>	
An Approximation Scheme for a Weighted Two-Cluster Partition Problem	323
<i>Alexander Kel'manov, Anna Motkova, and Vladimir Shenmaier</i>	
Hitting Set Problem for Axis-Parallel Squares Intersecting a Straight Line Is Polynomially Solvable for Any Fixed Range of Square Sizes	334
<i>Daniel Khachay, Michael Khachay, and Maria Poberiy</i>	
Polynomial Time Solvable Subclass of the Generalized Traveling Salesman Problem on Grid Clusters	346
<i>Michael Khachay and Katherine Neznakhina</i>	

Stabbing Line Segments with Disks: Complexity and Approximation Algorithms	356
<i>Konstantin Kobyakin</i>	
Analysis of Dynamic Behavior Through Event Data	
On the Efficient Application of Aho-Corasick Algorithm in Process Mining	371
<i>Andrey M. Konchagin and Anna A. Kalenkova</i>	
Social Network Analysis	
Health, Grades and Friendship: How Socially Constructed Characteristics Influence the Social Network Structure	381
<i>Sofia Dokuka, Ekaterina Krekhovets, and Margarita Priymak</i>	
Dynamic Semantic Network Analysis of Unstructured Text Corpora	392
<i>Alexander Kharlamov, Galina Gradoselskaya, and Sofia Dokuka</i>	
Scientific Matchmaker: Collaborator Recommender System	404
<i>Ilya Makarov, Oleg Bulanov, Olga Gerasimova, Natalia Meshcheryakova, Ilia Karpov, and Leonid E. Zhukov</i>	
Erratum to: Bagging Prediction for Censored Data: Application for Theatre Demand	E1
<i>Evgeniy M. Ozhegov and Alina Ozhegova</i>	
Author Index	411

Natural Language Processing

Automated Detection of Adverse Drug Reactions from Social Media Posts with Machine Learning

Ilseyar Alimova^(✉) and Elena Tutubalina

Laboratory of Chemoinformatics and Molecular Modeling,
Kazan (Volga Region) Federal University, Kazan, Russia
alimovailseyar@gmail.com, elvtutubalina@kpfu.ru

Abstract. Adverse drug reactions can have serious consequences for patients. Social media is a source of information useful for detecting previously unknown side effects from a drug since users publish valuable information about various aspects of their lives, including health care. Therefore, detection of adverse drug reactions from social media becomes one of the actual tools for pharmacovigilance. In this paper, we focus on identification of adverse drug reactions from user reviews and formulate this problem as a binary classification task. We developed a machine learning classifier with a set of features for resolving this problem. Our feature-rich classifier achieves significant improvements on a benchmark dataset over baseline approaches and convolutional neural networks.

Keywords: Adverse drug reactions · Text mining
Health social media analytics · Machine learning · Deep learning

1 Introduction

Detection of drug side effects is one of the main tasks in the pharmacy industry. Before the release of the medication, a number of clinical trials are conducted in order to detect side effects, which further fit into the drug's instructions. However, clinical trials do not allow to identify all drug side effects, because some of them appear after long-term use of the drug or have an effect only on a certain group of patients who did not participate in clinical trials. Recent work have shown that 462 medical products were withdrawn from sales between 1950 and 2014 years [1]. Moreover, side effects appeared in the post-approval period can cause serious problems to human health and even lead to death [2–5]. The detection of drug side effects in post-approval periods is a difficult challenge for pharmacovigilance.

One of the methods of finding new side effects for released into the market drugs is social media analysis [6]. With the development of social networks, users often write about the problems associated with taking medications on Twitter and various forums related to health and drugs. Manual processing such

a volume of text information is impossible, therefore methods of natural language processing are widely used for automatic text processing [7–10] including to extract information about adverse effects, as evidenced by a number of review articles on this topic [11–15].

One of the tasks of detection of Adverse Drug Reactions (ADR) is to classify any disease-related information. This classification task is necessary to remove noise information and detect if the text contains mentions of side effects. We formulated the problem as a binary classification to determine whether this side effect is adverse or not. The first class includes all the drug effects that had a negative impact on a health. The other group includes indications, disease symptoms, beneficial effects and effects experienced not by the patient directly. We developed a machine learning classifier with a set of features for resolving this problem. We used word embedding features, including vector representation and brown clusters trained on social media posts on the topic of health and drugs. We will demonstrate the effectiveness of our approach on message-level and entity-level classification. For message-level classification, we tested our approach on a benchmark dataset of tweets named the Twitter corpus [16]. For entity-level, we tested our approach on a benchmark dataset of user reviews named the CSIRO Adverse Drug Event Corpus (CADEC) [17]. We used Logistic Regression and Linear Support Vector Machine (SVM) classifiers. Our feature-rich classifiers outperformed Convolutional Neural Networks (CNN) designed for text classification [18] and a state-of-the-art approach based on SVM introduced by Sarker et al. [16].

The rest of the paper is structured as follows. In Sect. 2, we discuss related work. In Sect. 3, we describe our classifier with hand-crafted features. Section 4 provides evaluation results. Section 5 concludes this paper.

2 Related Work

Many previous works have been devoted to the detection of ADR. In Social Media Mining Shared Task 2016 [19], the task 1 was to classify user posts whether a user discusses ADRs. Participants were given a marked body of tweets on the topic of health, their task was to determine whether there was information about side effects in the text of the tweet or not. The winner system [20] is based on Random Forest and used words, the co-occurrence of drug and side effect, sentiment, negation, change in tweet’s tone and question in tweets features and obtained ADR class F-score of 41.95%. The second place system used different configurations of Maximum Entropy Classifier and concept-matching classifier based on ADR lexicon and obtained ADR class F-measure of 41.82% [21]. Ofoghi et al. [22] introduced SVM-based classifier with a sentiment, emotion classes, mention from Unified Medical Language System (UMLS), chemicals/drugs/diseases lexicon based features and got ADR class F-measure of 35.8%. The fourth-placed system applied SVM classifier with syntactic, lexicon, polarity and topic modeling based features and performed ADR class F-measure of 33% [23]. The last system from top five also developed the SVM-based system with n-Gram, word

embedding, cluster, lexical features and obtained F-measure of 31.74% [24]. SVM is the most popular text classification techniques, which has been widely adopted in patient social media research [25–28]. Sarker and Gonzalez [16] tried Maximum Entropy model with n-gram, UMLS semantic types and concept IDs, syn-set expansion, change phrases, ADR lexicon matches, SentiWordNet scores, topic-based features and obtained ADR class F-measures of 81.2%, 53.8% and 67.8% for Twitter¹, DailyStrength [28] and ADR corpora [29], respectively. Liu et al. [30] developed a rule-based algorithm to identify adverse drug events and event pairs from all related drugs. The classifier obtained macro-averaged F-measure of 69.20%. Huynh et al. [31] introduced CNN for classification of ADRs. CNN with pre-trained word embeddings showed better results over machine learning classifiers with macro-averaged F-measures of 51% and 87% on Twitter [16] and the second corpus of adverse events [32], respectively.

As you can see, basically, for the task ADR classification, machine-learning approaches were used with similar sets of features, including n-grams, ADR lexicon, sentiment features, a presence of drugs, UMLS mentions. Only in [24] were used embedding clusters features, trained with word2vec on unlabeled user reviews about drugs, collected from Twitter and Daily Strength, and k-means Clustering from [33].

3 The Proposed Method

We applied two machine learning models based on Linear SVM and Logistic Regression with entity-level and context-level sets of features. Entity-level features were applied only for entity tokens. Context-level features were used within a window of four words on each side of an entity, including the entity itself. For tweets from the Twitter corpus, we applied the features described below to all tokens in the tweet's text. The methods were implemented with classes from the scikit-learn library [34]: LinearSVC and Logistic Regression (with parameters `class_weight = 'auto'` and `penalty = 'l2'`). The source code of our classifier can be found here².

3.1 Features

Context-level features:

- Bag of words (bow): we used unigrams and bigrams.
- Part of Speech (PoS): we counted the number of nouns, verbs, adverbs and adjectives.
- Sentiment features (sent): we applied state-of-the-art lexicon features for sentiment analysis described in [35]. We used SentiWordNet [36], MPQA Subjectivity Lexicon [37], Bing Liu's dictionaries [38].

¹ <http://diego.asu.edu/index.php?downloads=yes>.

² https://github.com/Ilseyar/adr_classification.

- Pointwise mutual information (pmi): we counted PMI for the large corpus of user reviews named the Health corpus collected from various resources and described further. We used the PMI score as a feature.
- Drugname and ADR presenting (drug_adr): this is a binary vector of length two. The first component of the vector shows the presence of the name of the drug from FDA³, the second is the presence of ADR effect from the dictionary COSTART⁴.
- Emoticons (emot): a binary vector of length 2, showing the presence of positive and negative emoticons.

Entity-level features:

- Word embedding (emb): we used vector representation trained on social media posts from [39]. We calculated the average of the vectors of dimension 200 for each token and applied it as a feature.
- Cluster-based representation (cls): we used clusters computed in [39] with Brown hierarchical clustering algorithm and represented each entity as a binary vector of dimension 150.
- Semantic Types from Unified Medical Language System (umls): we used UMLS version 2.0 and counted the number of tokens from each UMLS semantic types. UMLS semantic types are subject categories that provide a categorization of all concepts represented in the UMLS. For example, clinical drug, medical device, vitamin etc.

Word embedding vectors were obtained with using word2vec trained on unlabeled Health corpus consists of 1,180,080 reviews from askapatient.com⁵, daily-strength.org⁶, drugscom.com⁷, amazon health-related dataset⁸, webmd.com⁹. The parameters of word embeddings are vectors with size of 200, the length of the local context of 10, the negative sampling of 5, vocabulary cutoff of 10, Continuous Bag of Words model [39].

4 Experiments

In this section, we describe our experiments with feature-rich classifiers and deep learning models.

4.1 Evaluation Datasets

We conducted experiments on CADEC [17] and Twitter corpora [16].

³ <https://www.fda.gov/>.

⁴ <http://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/CST/>.

⁵ <http://www.askapatient.com/>.

⁶ <https://www.dailystrength.org/>.

⁷ <https://www.drugs.com/>.

⁸ <http://jmcauley.ucsd.edu/data/amazon/>.

⁹ <http://www.webmd.com/>.

CADEC. The CADEC corpus consists of annotated user reviews from the ask-a-patient.com medical forum. There are five types of annotations: drug, adverse effect, disease, symptom, and finding. The ‘drug’ label was given to all drug names in the text. All side effects that are directly related to the drug, which was written about in the review, are annotated as ‘ADR’. The entity ‘disease’ is the indication to taking the drug. The entity labeled as ‘symptom’ specifies the indication disease. The label ‘finding’ was given to any side effects or symptoms, which are not related to the patient, and for entities that annotators could not establish belonging to one of the classes. We grouped diseases, symptoms and findings as single class called ‘Other’. The corpus contains 6320 entities, 5770 of them marked as ‘ADR’.

Twitter. The Twitter corpus contains user tweets on the topic of health and adopted from [16]. Each tweet labeled whether tweet’s text contains the information about adverse drug reactions. Since the policy of Twitter does not allow the publication of tweet texts in the public domain, the corpus consists of a file containing the id tweet, the user id, and the class number. The creators of the corpus published a script for downloading tweet texts. A fraction of the tweets (36%) was no longer available on Twitter, which made our results are not directly comparable to the ones of previous works. During pre-processing, we removed all URLs, user mentions and symbols of re-tweets using the tweet-preprocessor package¹⁰.

4.2 Baseline Methods

We compare our approach with two baselines:

- **SVM from [16]:** The developed method based on SVM with Linear kernel. Features applied in this method incorporated 1, 2, 3-g, synsets, sentiment, change phrases, ADR lexicon, topic-based features, the lengths of the text segments in words, the presence of comparatives and superlatives adjectives and modal verbs. The synsets features consist of synonyms for each adjective, noun or verb in a sentence obtained from WordNet. For sentiment feature, the following dictionaries were used: SentiWordNet [36], MPQA Subjectivity Lexicon [37], Bing Liu’s dictionary [38]. The change phrases feature presented a vector of dimension 4, where each component shows the number of words belonging to ‘less’, ‘more’, ‘good’, ‘bad’ words dictionaries. ADR lexicon feature used SIDER¹¹, Consumer Health Vocabulary¹², COSTART¹³ and DIEGO_LAB¹⁴ dictionaries and consists of two parameters, the first shows the token belonging to the ADR dictionary, the second number of tokens from the ADR dictionary. The topic-based feature consists of the topic terms that

¹⁰ <https://pypi.python.org/pypi/tweet-preprocessor/0.4.0>.

¹¹ <http://sideeffects.embl.de/>.

¹² <http://www.consumerhealthvocab.org/>.

¹³ <https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/CST/>.

¹⁴ <http://diego.asu.edu/Publications/ADRClassify.html>.

appear in the instance and the sums of all the relevance scores of the terms in each instance. We used the publicly available code of this method¹⁵.

- **CNN:** In order to get local features from a review with CNNs we have used multiple filters of different lengths [18]. Pooled features are fed to a fully connected feed-forward neural network (with dimension 100) to make an inference, using rectified linear units as output activation. Then we apply a softmax classifier with a number of outputs equals 2. We applied dropout rate of 0.5 [40] to the fully connected layer. We trained CNN for 10 epochs since CNN achieved lower results after ten epochs. Embedding layers are trainable for all networks; this setting leads to a significant gain in performance. We set mini-batch size to 128 with the Adam optimizer [41]. We found 97% and 84% of words in the vocabulary in the CADEC corpus and the Tweet corpus, respectively, and for other words, the representations were uniformly sampled from the range of embedding weights [42]. We used the Keras library [43] for implementation. Both CNN and our classifier used the same word embeddings trained on health-related comments from [39].

4.3 Results and Analysis

We performed pre-processing by lower-casing all words. We tested the methods on the 5-folds cross validation. We computed macro-averaged recall (R), precision (P) and F₁-measures (F). Table 1 presents the variances of the F1-measure in our cross-validation results. We took the best configuration of features for each model, bow, pos, sent, cls, umls for Linear SVM and all set of described features for Logistic Regression model.

The classification results (especially F1-measure) in Table 1 indicate the advantage of machine learning classifiers. The feature-rich SVM and Logistic Regression classifiers achieved best results on the CADEC corpus and the Twitter corpus, respectively. Therefore, classical machine learning approaches with rich additional information can still outperform neural network approaches for domain-specific problems like detection of adverse drug reactions.

We also investigated the effectiveness of features in Table 2. As can be seen from the tables, Linear SVM with features obtained the maximum value of the macro F-measure 80.3% on CADEC corpus and Logistic Regression got macro F-measure 73.7% on the Twitter corpus. For SVM, cluster-based features and umls increase significantly the effectiveness of the classifier on the CADEC corpus. For Logistic Regression, most significant feature is word embeddings.

We also investigated the effectiveness of different sentiment lexicons. The sentiment feature was computed only for the Bing Liu's lexicon since the full set of sentiment lexicons didn't improve the performance. We tried to extend our set of features with ADR lexicons similar to [16]. We used COSTART¹⁶, SIDER¹⁷ and DIEGO Lab ADR Lexicon from [16] with the best configurations of features.

¹⁵ <https://bitbucket.org/asarker/adrbinaryclassifier/downloads/>.

¹⁶ <http://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/CST/>.

¹⁷ <http://sideeffects.embl.de/>.

Table 1. 5-fold cross-validation performances of our feature-rich methods and baselines.

Method	Corpus	Folds (F ₁ -measure)					av. F
		1	2	3	4	5	
SVM + best config.	CADEC	.783	.788	.817	.783	.846	.803
Logistic Regression + best config.	CADEC	.787	.767	.791	.785	.782	.783
CNN, [1, 2] filters	CADEC	.760	.771	.792	.805	.784	.782
CNN, [1, 2, 3] filters	CADEC	.740	.760	.770	.797	.777	.769
CNN, [2, 3, 4] filters	CADEC	.771	.739	.799	.787	.757	.771
CNN, [1, 2, 3, 4] filters	CADEC	.765	.766	.773	.804	.785	.779
CNN, [1, 2, 3, 4, 5] filters	CADEC	.796	.773	.768	.794	.787	.783
Classifier from [16]	CADEC	.645	.676	.737	.677	.703	.688
SVM + best config.	Twitter	.683	.715	.706	.700	.709	.702
Logistic Regression + best config.	Twitter	.711	.752	.747	.737	.738	.737
CNN, [1, 2] filters	Twitter	.628	.735	.739	.714	.695	.702
CNN, [1, 2, 3] filters	Twitter	.635	.716	.726	.708	.725	.702
CNN, [2, 3, 4] filters	Twitter	.692	.703	.722	.712	.651	.696
CNN, [1, 2, 3, 4] filters	Twitter	.695	.705	.749	.674	.642	.693
CNN, [1, 2, 3, 4, 5] filters	Twitter	.677	.701	.725	.694	.693	.698
Classifier from [16]	Twitter	.676	.677	.691	.680	.698	.684

Table 2. Features impact evaluation on the CADEC and Twitter corpora with Linear SVM and Logistic Regression models respectively with different groups of features.

Features	CADEC			Twitter		
	P	R	F	P	R	F
bow	.827	.740	.775	.642	.751	.666
bow, pos	.824	.743	.776	.722	.682	.700
bow, pos, sent	.823	.745	.777	.719	.698	.708
bow, pos, sent, cls	.832	.776	.788	.721	.708	.714
bow, pos, sent, cls, umls	.844	.773	.803	.721	.708	.714
bow, pos, sent, cls, umls, pmi	.839	.770	.799	.723	.710	.716
bow, pos, sent, cls, umls, pmi, emb	.822	.777	.797	.730	.742	.736
bow, pos, sent, cls, umls, pmi, emb, drug_adr	.842	.772	.802	.728	.745	.736
bow, pos, sent, cls, umls, pmi, emb, drug_adr, emot	.812	.774	.792	.729	.746	.737

The results of these experiments are presented in Table 3. According to these, only COSTART lexicon improved classification results for Twitter corpus. The possible explanation is that the dictionaries provide poor coverage of the corpora, as shown in the Table 4.

Table 3. Lexicon features impact evaluation on the CADEC and Twitter corpora with Linear SVM and Logistic Regression models respectively with groups of features with the best results.

Lexicons	CADEC			Twitter		
	P	R	F	P	R	F
COSTART	.842	.767	.799	.729	.744	.736
SIDER	.840	.768	.799	.729	.739	.734
ADR lexicon	.844	.769	.801	.719	.698	.708

Table 4. Summary of statistics of considered lexicons

Corpus	unigrams	COSTART	SIDER	ADR lexicon
CADEC	3204	111	145	311
Twitter	11929	95	149	315

4.4 Error Analysis

In this section, we present an analysis of classification errors. We looked at 150 examples of each type of errors and identified the main causes of errors with the examples presented in the Table 5. The other cases are difficult to combine into large groups, each of them requires more detailed consideration.

CADEC Corpus

- *ADR with pain word.* Most errors are associated with the entities that include word ‘pain’ (30%). If the word ‘pain’ was next to the words denoting negative sentiment, the system erroneously classified it as ‘Adverse’, however, it becomes clear from the context that in this case, this is not an adverse drug reaction. On the contrary, if the entity ‘pain’ was encountered in a positive context, the system classified these cases as ‘Other’, but often in such cases, the patient described his condition after he stopped taking the medicine and the annotators labeled it as ‘Other’.
- *Training set imbalance.* Some errors appeared because of the imbalance in the training set. For example, the entities: ‘depression’, ‘swelling’, ‘headache’, ‘cramps’ most often belong to the class ‘Adverse’, and the words ‘inflammation’, ‘fibromyalgia’, ‘MS’ are mostly classified as ‘Other’. The error associated with this is more common for the case where the system incorrectly labeled entities as Adverse (about 18%) and for another type of errors it is only 5%.
- *ADR context.* The presence of terms denoting the adverse drug reaction next to the entity, caused the system to erroneously give the answer ‘Adverse’ instead of right ‘Other’. It caused 25% of errors.

Table 5. Examples illustrating common reasons behind the misclassification

Review's text	Corpus	Classified as	Issues
If I miss a day, headaches begin to creep in	CADEC	Adverse	Training set disbalance
...depression worse, fibromyalgia much worse	CADEC	Other	Training set disbalance
your tweets are depressing me. haha. paxil	Twitter	Adverse	Adverse drug context
...by an adverse reaction to the drug effexor	Twitter	Adverse	No ADR description
...12 rivaroxaban diary: headache, right shoulder...	Twitter	Adverse	Not own user experience
I've had no appetite since i started on prozac, i guess that's a good thing	Twitter	Adverse	Positive side effect
Nicotine lozenges. if i go cold turkey i can't think (or see) straight...	Twitter	Adverse	No drug name
...feel like this fluoxetine is messing with my perspective time	Twitter	Other	No ADR mention

Twitter Corpus

- *No ADR mention.* The absence of mention of the concrete adverse effect is the most common reason of the erroneous decision of our system to label the tweet as ‘Other’. This is 40% of all cases. Sometimes such cases describe a patient’s violation of a diet, in particular, alcohol consumption during the drug taking, or overdose.
- *Adverse drug context.* The second most common error (about 40%) is associated with the mention of the drug and the side effects together in the text of the tweet. While the system determined it to the ‘Adverse’ class, the right answer was ‘Other’ because the effects did not apply to the drug.
- *No ADR description.* In 40% of erroneously classified tweets, users wrote about the presence of ADR but did not describe them specifically. In this case, the system classified this tweet as ‘Adverse’, however, in the gold file such cases was annotated as ‘Other’.
- *Not own user experience.* In 15% of cases, the user described the side effect of the drug that he did not experience, in this cases our system also recognized this tweet as ‘Adverse’, although, the correct answer was ‘Other’.
- *Positive side effect.* If the adverse effect was identified as positive by user the system defined the tweet in the category ‘Other’ class, however in the gold file it was labeled as ‘Adverse’ (10%).
- *No drug name.* The lack of mention of the drug name also led to an error when the system incorrectly classified a tweet to ‘Other’ (10%).

5 Conclusion

In this paper, we have focused on the ADR classification task. We have explored Linear SVM and Logistic Regression classifiers with a rich set of features including sentiment and semantic features, word embeddings, and lexicon features. We tested the proposed approach on two benchmark corpora of user reviews and tweets and compared with the state-of-the-art classifier and convolutional neural networks.

We demonstrated the superiority of machine learning approach as compared to a convolutional neural network and another previously proposed approach. The most improvement for ADR classification gave sentiment and word embedding features. We also showed that the features based on ADR lexicon do not give significantly improve for classification results. The possible explanation is that a dictionary of ADRs is specific for a particular group of drugs (e.g., weight gain or loss is a side effect of depression treatment and the reason for taking orexigenic drugs and appetite suppressants). Hence, in further work, we plan to create drug-specific dictionaries and incorporate them into neural models.

Acknowledgments. Work on problem definition and neural networks was carried out by Elena Tutubalina and supported by the Russian Science Foundation grant no. 15-11-10019. Other parts of this work were performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

References

1. Onakpoya, I.J., Heneghan, C.J., Aronson, J.K.: Post-marketing withdrawal of 462 medicinal products because of adverse drug reactions: a systematic review of the world literature. *BMC Med.* **14**(1), 10 (2016)
2. Pirmohamed, M., James, S., Meakin, S., Green, C., Scott, A.K., Walley, T.J., Farrar, K., Park, B.K., Breckenridge, A.M.: Adverse drug reactions as cause of admission to hospital: prospective analysis of 18 820 patients. *BMJ* **329**(7456), 15–19 (2004)
3. Classen, D.C., Pestotnik, S.L., Evans, R.S., Lloyd, J.F., Burke, J.P.: Adverse drug events in hospitalized patients: excess length of stay, extra costs, and attributable mortality. *JAMA* **277**(4), 301–306 (1997)
4. Lazarou, J., Pomeranz, B.H., Corey, P.N.: Incidence of adverse drug reactions in hospitalized patients: a meta-analysis of prospective studies. *JAMA* **279**(15), 1200–1205 (1998)
5. Bates, D.W., Cullen, D.J., Laird, N., Petersen, L.A., Small, S.D., Servi, D., Laffel, G., Sweitzer, B.J., Shea, B.F., Hallisey, R., et al.: Incidence of adverse drug events and potential adverse drug events: implications for prevention. *JAMA* **274**(1), 29–34 (1995)
6. Sloane, R., Osanlou, O., Lewis, D., Bollegala, D., Maskell, S., Pirmohamed, M.: Social media and pharmacovigilance: a review of the opportunities and challenges. *Br. J. Clin. Pharmacol.* **80**(4), 910–920 (2015)
7. Tutubalina, E., Nikolenko, S.: Automated prediction of demographic information from medical user reviews. In: Prasath, R., Gelbukh, A. (eds.) MIKE 2016. LNCS, vol. 10089, pp. 174–184. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58130-9_17

8. Solovyev, V., Ivanov, V.: Knowledge-driven event extraction in Russian: corpus-based linguistic resources. *Comput. Intell. Neurosci.* **2016**, 16 (2016)
9. Sayfullina, L., Eirola, E., Komashinsky, D., Palumbo, P., Karhunen, J.: Android malware detection: building useful representations. In: 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 201–206, December 2016
10. Ivanov, V., Tutubalina, E., Mingazov, N., Alimova, I.: Extracting aspects, sentiment and categories of aspects in user reviews about restaurants and cars. In: Proceedings of International Conference Dialog, vol. 2, pp. 22–34 (2015)
11. Murff, H.J., Patel, V.L., Hripcak, G., Bates, D.W.: Detecting adverse events for patient safety research: a review of current methodologies. *J. Biomed. Inform.* **36**(1), 131–143 (2003)
12. Sarker, A., Ginn, R., Nikfarjam, A., O'Connor, K., Smith, K., Jayaraman, S., Upadhyaya, T., Gonzalez, G.: Utilizing social media data for pharmacovigilance: a review. *J. Biomed. Inform.* **54**, 202–212 (2015)
13. Lardon, J., Abdellaoui, R., Bellet, F., Asfari, H., Souvignet, J., Texier, N., Jaulent, M.C., Beyens, M.N., Burgun, A., Bousquet, C.: Adverse drug reaction identification and extraction in social media: a scoping review. *J. Med. Internet Res.* **17**(7), e171 (2015)
14. Harpaz, R., Callahan, A., Tamang, S., Low, Y., Odgers, D., Finlayson, S., Jung, K., LePendu, P., Shah, N.H.: Text mining for adverse drug events: the promise, challenges, and state of the art. *Drug Saf.* **37**(10), 777–790 (2014)
15. Harpaz, R., DuMouchel, W., Shah, N.H., Madigan, D., Ryan, P., Friedman, C.: Novel data-mining methodologies for adverse drug event discovery and analysis. *Clin. Pharmacol. Ther.* **91**(6), 1010–1021 (2012)
16. Sarker, A., Gonzalez, G.: Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *J. Biomed. Inform.* **53**, 196–207 (2015)
17. Karimi, S., Metke-Jimenez, A., Kemp, M., Wang, C.: Cadec: a corpus of adverse drug event annotations. *J. Biomed. Inform.* **55**, 73–81 (2015)
18. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint [arXiv:1408.5882](https://arxiv.org/abs/1408.5882) (2014)
19. Sarker, A., Nikfarjam, A., Gonzalez, G.: Social media mining shared task workshop. In: Proceedings of the Pacific Symposium on Biocomputing, pp. 581–592 (2016)
20. Rastegar-Mojarad, M., Komandur Elayavilli, R., Yu, Y., Hiu, H.: Detecting signals in noisy data-can ensemble classifiers help identify adverse drug reaction in tweets. In: Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing (2016)
21. Zhang, Z., Nie, J., Zhang, X.: An ensemble method for binary classification of adverse drug reactions from social media. In: Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing (2016)
22. Ofoghi, B., Siddiqui, S., Verspoor, K.: Read-BioMed-SS: adverse drug reaction classification of microblogs using emotional and conceptual enrichment. In: Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing (2016)
23. Jonnagaddala, J., Jue, T.R., Dai, H.: Binary classification of twitter posts for adverse drug reactions. In: Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing, pp. 4–8 (2016)
24. Egger, D., Uzdilli, F., Cieliebak, M., Derczynski, L.: Adverse drug reaction detection using an adapted sentiment classifier. In: Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing (2016)

25. Ginn, R., Pimpalkhute, P., Nikfarjam, A., Patki, A., O'Connor, K., Sarker, A., Smith, K., Gonzalez, G.: Mining twitter for adverse drug reaction mentions: a corpus and classification benchmark. In: Proceedings of the Fourth Workshop on Building and Evaluating Resources for Health and Biomedical Text Processing. Citeseer (2014)
26. Yang, M., Wang, X., Kiang, M.Y.: Identification of consumer adverse drug reaction messages on social media. In: PACIS, vol. 193 (2013)
27. Bian, J., Topaloglu, U., Yu, F.: Towards large-scale twitter mining for drug-related adverse events. In: Proceedings of the 2012 International Workshop on Smart Health and Wellbeing, pp. 25–32. ACM (2012)
28. Patki, A., Sarker, A., Pimpalkhute, P., Nikfarjam, A., Ginn, R., O'Connor, K., Smith, K., Gonzalez, G.: Mining adverse drug reaction signals from social media: going beyond extraction. In: Proceedings of BioLinkSig 2014, pp. 1–8 (2014)
29. Gurulingappa, H., Mateen-Rajpu, A., Toldo, L.: Extraction of potential adverse drug events from medical case reports. *J. Biomed. Semant.* **3**(1), 15 (2012)
30. Liu, X., Liu, J., Chen, H.: Identifying adverse drug events from health social media: a case study on heart disease discussion forums. In: Zheng, X., Zeng, D., Chen, H., Zhang, Y., Xing, C., Neill, D.B. (eds.) ICSH 2014. LNCS, vol. 8549, pp. 25–36. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08416-9_3
31. Huynh, T., He, Y., Willis, A., Rüger, S.: Adverse drug reaction classification with deep neural networks. In: COLING (2016)
32. Gurulingappa, H., Rajput, A.M., Roberts, A., Fluck, J., Hofmann-Apitius, M., Toldo, L.: Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *J. Biomed. Inform.* **45**(5), 885–892 (2012)
33. Nikfarjam, A., Sarker, A., O'Connor, K., Ginn, R., Gonzalez, G.: Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *J. Am. Med. Inform. Assoc.* **22**(3), 671–681 (2015)
34. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
35. Kiritchenko, S., Zhu, X., Mohammad, S.M.: Sentiment analysis of short informal texts. *J. Artif. Intell. Res.* **50**, 723–762 (2014)
36. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In: LREC, vol. 10, pp. 2200–2204 (2010)
37. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 347–354. Association for Computational Linguistics (2005)
38. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177. ACM (2004)
39. Miftahutdinov, Z., Tutubalina, E., Tropsha, A.: Identifying disease-related expressions in reviews using conditional random fields. *Komp'yuternaja Lingvistika i Intellektual'nye Tehnologii* **1**(16), 155–166 (2017)
40. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)

41. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
42. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034 (2015)
43. Chollet, F., et al.: Keras (2015). <https://github.com/fchollet/keras>

Automated Detection of Non-Relevant Posts on the Russian Imageboard “2ch”: Importance of the Choice of Word Representations

Amir Bakarov^{1,2(✉)} and Olga Gureenkova^{1,2}

¹ Chatme AI LLC, ul. Nikolaeva 11, of. 707, Novosibirsk, Russia

{a.bakarov, o.gureenkova}@expasoft.ru

² Novosibirsk State University, Novosibirsk, Russia

Abstract. This study considers the problem of automated detection of non-relevant posts on Web forums and discusses the approach of resolving this problem by approximating it with the task of detection of semantic relatedness between the given post and the opening post of the forum discussion thread. The approximated task could be resolved through learning the supervised classifier with a composed word embeddings of two posts. Considering that the success in this task could be quite sensitive to the choice of word representations, we propose a comparison of the performance of different word embedding models. We train 7 models (Word2Vec, Glove, Word2Vec-f, Wang2Vec, AdaGram, FastText, Swivel), evaluate embeddings produced by them on dataset of human judgements and compare their performance on the task of non-relevant posts detection. To make the comparison, we propose a dataset of semantic relatedness with posts from one of the most popular Russian Web forums, imageboard “2ch”, which has challenging lexical and grammatical features.

Keywords: Distributional semantics · Compositional semantics
2ch · Imageboard · Semantic relatedness · Word similarity
Word embeddings

1 Introduction

Currently many of the Web forums work not only as platforms for conversational entertainment but also as free sources of information in different domains of human knowledge. However, these sources are becoming significantly noised with large amounts of non-relevant posts like flame, cyber-bullying, political provocations or any other types of posts that obstruct productive discussions and interrupt convenient reading of forum thread; so, *non-relevant posts could be considered as not related to the topic of the opening post of the Web forum discussion thread*. Therefore, the task of automated detection of non-relevant posts, the solution of which will allow simplifying the process of their deletion,

could be approximated with the task of automated detection of *semantic relatedness* (which, in this study, we consider as an existence of a common concept or a field between two linguistic units) between the given post and the opening post of the forum discussion thread.

In recent research studies the task of semantic relatedness detection is usually resolved by modeling semantic meaning of the matched linguistic units. This modeling is usually performed with the help of *distributional semantic models*, the approaches that can represent linguistic units through dense real-valued vectors, and if the units are words, the vectors will be called *word embeddings (WE)*. However, we believe that the success of such modeling is quite sensitive to the choice of word representations which vary with different word embedding models (WEM). To this end, we propose a comparison of the models in the task of semantic relatedness detection based on the approach when the vector of a posts pair could be obtained as an arithmetic mean of their non-ordered WE. So, we propose a dataset for semantic relatedness with posts from one of the most popular Russian Web forums, *imageboard “2ch”* (“Двач”; <https://2ch.hk/>), containing a lot of Web slang vocabulary, misspellings, typos and abnormal grammar. But, firstly, we evaluate different WE on the dataset of word similarities to ensure that the single word representation proposed by the compared models are really different. To summarize, our main contributions are the following:

- Our work is the first towards a survey of the WEM applied to the textual data of Russian language, and we suggest a model for automated detection of non-relevant Web forum posts which obtained a maximum F1-score of 0.85 on our data;
- We provide a manually annotated Russian Web slang dataset of semantic relatedness containing 2663 post pairs.

The paper is organized as follows. Section 2 provides a survey on the related work in the given task. In Sect. 3 we provide description of our dataset. In Sect. 4 the details of the experiments are described. Section 5 covers the results of the comparison and Sect. 6 concludes the work.

2 Related Work

In recent years the research interest to online social media have significantly increased, and different studies have explored the Web forums from the point of natural language processing tasks like speech acts classifying [1]. However, we are not aware of any research in the task of post relevance detection, especially from the perspective of semantic relatedness detection of complex linguistic units (like sentences and texts) for the Russian language. But the detection of semantic relatedness itself has a broad amount of resolutions proposed by other researchers; the extensive survey of them is presented at the official web-page of *Stanford Natural Language Inference Corpus* (<https://nlp.stanford.edu/projects/snli/>). For English most of the research of semantic relatedness/similarity were proposed as the part of *SEM shared tasks (for example,

for the task of textual semantic similarity detection [2]); there were also some studies in the task of word similarity for Russian language as a part of *RuSSE* [3]. In this study we will also use datasets proposed on RuSSE to evaluate the word representations obtained with different WEM.

3 Dataset for Semantic Relatedness

To propose the comparison in the task of semantic relatedness detection we created a dataset of 2663 Russian language pairs of short (up to 216 symbols) texts based on a set of posts mined from 45 different discussion threads of “2ch” (*2ch Semantic Relatedness Dataset, 2SR*). The dataset is presented in a form of a list of triples (*post*, *op_post*, *is_related*) (which stands for “single post”, “opening post”, “existence of semantic relatedness”) and contains human judgements about existence of semantical relatedness between the given post and the opening post in a form of binary labels; the distribution of labels in the dataset is 48% to 52%. 2SR notably contains a large amount of duplicates of *op_post* since a single opening post is associated to a large amount of posts in the structure of the Web forums, and, due to the peculiarities of the source, it is filled with misspelled, slang and obscene vocabulary.

In order to collect the human judgements, three native speaking volunteers from Novosibirsk State University were invited to participate in the experiment. Each annotator was provided with the whole dataset and asked to assess the binary label to each pair choosing from options “relatedness exists” and “relatedness does not exist”. To conclude the inter-annotator agreement the final label for each pair was obtained as a label marked by most of the annotators.

4 Experimental Setup

4.1 Explored Models

For training WEM we created a corpus of 1 906 120 posts (614 707 unique words) from “2ch”. The downloaded posts were cleared from HTML-tags, hyperlinks and non-alphabetic symbols; we also lemmatized them with *pymorphy2*.

We set the dimensionality of word vectors to 100 (since it showed the better performance on our data across other dimensionalities); for every model we also picked the most efficient architecture based on the evaluation on our data. As a result, the following models were compared:

- **Word2Vec (CBOW)** [4]. Computation of the prediction loss of the target words from the context words. Used *gensim* implementation.
- **GloVe¹** [5]. Dimensionality reduction on the co-occurrence matrix.
- **Word2Vec-f (CBOW)²** [6]. Extension of Word2Vec with the use of arbitrary context features of dependency parsing³.

¹ <https://github.com/stanfordnlp/glove>.

² <https://bitbucket.org/yoavgo/word2vecf>.

³ This model was trained on a raw corpus represented in *CONLL-U* format through the parsing of *SyntaxNet Parsey McParseface* trained on *SynTagRus*.

- **Wang2Vec (Structured Skip-N-Gram)**⁴ [7]. Extension of Word2Vec with the sensitivity to the word order.
- **AdaGram**⁵ [8]. Extension of Word2Vec learning multiple word representations with capturing different word meanings⁶.
- **FastText (CBOW)**⁷ [9]. Extension of Word2Vec which represents words as bags of character n-grams.
- **Swivel** [10]. Capturing unobserved (word, context) pairs in sub-matrices of a co-occurrence matrix. Used *Tensorflow* implementation.

4.2 Word Semantic Similarity

First of all, we considered a comparison on the task of semantic similarity on three datasets of RuSSE: *HJ*, *RT* (test chunk) and *AE* (test chunk). For each word pair of the dataset we computed the cosine distance of the embeddings associated to them, and then calculated a Spearman’s correlation p and an average precision score (AP) between given cosine distances and human judgements. Entries containing at least one out-of-vocabulary (OOV) word were dropped, and amount of dropped pairs consisted 27.4% for the first dataset, 74.0% for the second and 38.0% for the third for Word2Vec-f models (since it uses a different vocabulary) and 5.5%/40.9%/9.4% for other models.

4.3 Semantic Relatedness of Short Texts

Secondly, for performing a comparison on the task of semantic relatedness detection, we transformed 2SR to a vector space. To obtain a single post vector, we associated a WE to each word in a single post (OOV words were not taken into account) and calculated the arithmetic mean of the obtained unordered vectors. Then, to obtain the “final vector” of the pair of posts we considered three possible ways:

- Arithmetic mean of the single posts vectors (SUM);
- Concatenation of the single posts vectors (CON);
- Concatenation and then reducing the dimensionality twice (we used a method of *Principal Component Analysis (PCA)*).

Then these “final vectors” were used as the feature matrix for learning the classifier, and the vector of labels `is_related` of 2SR was used as a target vector. The matrix and the vector were used as training data for the classifier which was implemented with K-Nearest Neighbors algorithm (KNN) with 3 folds and a cosine metric with the help of *scikit-learn* (we also tried to use other classification algorithms and obtained lower results). We used cross-validation on the training set by 10 folds to train and to evaluate KNN.

⁴ <https://github.com/wlin12/wang2vec>.

⁵ <https://github.com/lopuhin/python-adagram>.

⁶ Since AdaGram has an opportunity to predict multiple meanings for a single word, we used the most probable predicted meaning of 2 prototypes.

⁷ <https://github.com/facebookresearch/fastText>.

The code on Python 3.5.4, 2SR, training corpus and links to the models for reproducing the experiments are available on our GitHub: <https://github.com/bakarov/2ch2vec>.

5 Results

The results of the comparison on two tasks are proposed at the Table 1, and we also created plots of the learning curves of compared models which are proposed at the Fig. 1 to illustrate the process of training. The difference in values obtained for semantic similarity demonstrates that the word representations of the compared models disagree (for instance, the cosine distance between words “кошка” (cat) and “собака” (dog) was 0.74 for FastText model and 0.62 for GloVe model), since the models use different features of textual data to create the embeddings. And the difference in the embeddings leads to different performance in the task of semantic relatedness detection: the maximum interval between the scores reaches 0.05 of F1 which we consider as significant. So, Swivel and FastText are the best models for word similarity tasks, and Wang2Vec, Swivel and FastText are the best models for the semantic relatedness task.

Table 1. Performance of the vectors of the compared models across different tasks. Word similarity task reports Spearman’s p and AP with human annotation; semantic relatedness task reports $F1$ on different approaches of vector composing. In all cases, larger numbers indicate better performance.

Model	Semantic Similarity			Semantic Relatedness, $F1$		
	HJ, p	RT, AP	AE, AP	SUM	CON	CON+PCA
Word2Vec	0.51	0.72	0.78	0.836	0.852	0.831
GloVe	0.4	0.74	0.77	0.834	0.847	0.831
Word2Vec-f	0.04	0.73	0.74	0.782	0.787	0.809
Wang2Vec	0.41	0.72	0.78	0.839	0.85	0.84
AdaGram	0.11	0.57	0.66	0.8	0.819	0.79
FastText	0.44	0.76	0.79	0.832	0.854	0.841
Swivel	0.52	0.74	0.76	0.839	0.851	0.842

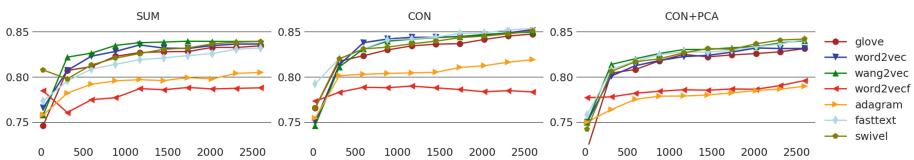


Fig. 1. Learning curves of KNN on cross-validation with three different options of vector composing with F1-score on Y-axis and an amount of training data on X-axis.

6 Conclusion

The considered experiments which were illustrated with the suggested algorithm for filtering the Web forum posts confirm our hypothesis that different word representations propose different results since the nature of their embeddings varies. The best F1 on the semantic relatedness task was achieved by FastText and CON method of obtaining the pair of posts vector. However, the same model trained on 2ch corpus did not perform so good in the task of semantic similarity. It can be concluded that not only the inner algorithm of a particular WEM affects the result, but also a vocabulary of a chosen corpus. Hypothetically, the best result on a corpus should be achieved by a model which inner algorithm better reflects the human perception of semantic relatedness between different words in the particular context of this vocabulary. In future we plan to extend the comparison on the Web forums of other languages and propose a typological comparison of semantic drifts of the Web slang meanings in different cultures illustrating it with word representations of different languages.

References

1. Qadir, A., Riloff, E.: Classifying sentences as speech acts in message board posts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 748–758 (2011)
2. Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W.: sem 2013 shared task: semantic textual similarity, including a pilot on typed-similarity. In: *SEM 2013: The Second Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics, Citeseer (2013)
3. Panchenko, A., Ustalov, D., Arefyev, N., Paperno, D., Konstantinova, N., Loukachevitch, N., Biemann, C.: Human and machine judgements for Russian semantic relatedness. In: Ignatov, D.I., et al. (eds.) AIST 2016. CCIS, vol. 661, pp. 221–235. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-52920-2_21
4. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111–3119 (2013)
5. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP, vol. 14, pp. 1532–1543 (2014)
6. Levy, O., Goldberg, Y.: Dependency-based word embeddings. In: ACL, vol. 2, pp. 302–308 (2014)
7. Ling, W., Dyer, C., Black, A.W., Trancoso, I.: Two/too simple adaptations of word2vec for syntax problems. In: HLT-NAACL, pp. 1299–1304 (2015)
8. Bartunov, S., Kondrashkin, D., Osokin, A., Vetrov, D.: Breaking sticks and ambiguities with adaptive skip-gram. In: Artificial Intelligence and Statistics, pp. 130–138 (2016)
9. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint [arXiv:1607.04606](https://arxiv.org/abs/1607.04606) (2016)
10. Shazeer, N., Doherty, R., Evans, C., Waterson, C.: Swivel: improving embeddings by noticing what's missing. arXiv preprint [arXiv:1602.02215](https://arxiv.org/abs/1602.02215) (2016)

A Morphological Processor for Russian with Extended Functionality

Elena I. Bolshakova^{1,2(✉)} and Alexander S. Sapin¹

¹ Lomonosov Moscow State University, Moscow, Russia
eibolshakova@gmail.com, alesapin@gmail.com

² National Research University Higher School of Economics, Moscow, Russia

Abstract. The paper presents an open-source morphological processor of Russian texts recently developed and named CrossMorphy. The processor performs lemmatization, morphological tagging of both dictionary and non-dictionary words, contextual and non-contextual morphological disambiguation, generation of word forms, as well as morphemic parsing of words. Besides the extended functionality, emphasis is put on linguistic quality of word processing and easy integration into programming projects. CrossMorphy is fully implemented in C++ programming language on the base of OpenCorpora vocabulary data. To clarify the reasons of its development, a comparison of several freely available morphological processors for Russian is given, across their linguistic and some technological properties. The experimental evaluation shows that CrossMorphy ensures rather high quality of word processing.

Keywords: Morphological tagging
Morphological parsers for Russian
Functionality of morphological processors
Morphological disambiguation · Morphemic parsing

1 Introduction

Morphological analysis of texts is a traditional task of computational linguistics and natural language processing (NLP). Almost any NLP system needs lemmatization and morphological tagging of word forms. For Russian language, methods of formal description of Russian morphology have been long known, and main problems of automatic morphological analysis are considered principally solved. However, some attendant and related problems are not fully solved, in particular, automatic morphemic analysis and morphological disambiguation. The latter is more complicated for languages with rich morphologies, such as Russian, and needs to be further investigated, since rather few topical works are known [1, 9–12, 14].

Nowadays, more than a dozen morphological processors for Russian are known, including freely available ones: AOT¹, Mystem², TreeTagger³, Pymorphy2⁴. The processors differ in their functionality and also in technological features. Most processors are appropriate for majority of NLP researches and applications that do not require any deep analysis of texts (e.g., categorization of texts). Nevertheless, for more complicated applied tasks (such as information extraction or question answering), morphological parsers with a specific combination of properties are needed. From this point of view, the set of freely available morphological processors for Russian is not complete.

In our research project on lexico-syntactic patterns language [4] intended to build various NLP applications on the basis of surface syntactic analysis, we used open source processor AOT¹. Unfortunately, it is not supported now and has some weak spots. The rest freely available morphological parsers are also not suitable: we lack a processor with the particular functionality and at the same time with the ability to integrate it in our project. So we were forced to begin development of our own morphologic processor, with emphasis to extended functionality, linguistic quality of word processing, and easy integration into programming projects. We suppose that yet another open source module with the particular linguistic and technological properties will be useful not only for us, and our efforts are a step towards collection of high-quality open-source morphological tools for Russian useful for various applications.

In this paper we present the developed morphological processor CrossMorphy⁵ that is open source software fully implemented in C++ language and based on freely available data of Open Corpora⁶. To clarify CrossMorphy's peculiarities and reasons of its development, we begin with a comparison of the most used and freely available morphological parsers for Russian. We consider their properties including pure linguistic (such as lemmatization, morphological tagging, generation of word forms) and also some technical features important for our purposes (such as ability to integrate source code of processor to NLP programming project or to connect a specific dictionary). Then we explain main decisions undertaken while developing CrossMorphy and describe its functionality, which encompasses, besides main linguistic properties, morphological disambiguation and morphemic parsing. Evaluation of the described CrossMorphy's functionality shows sufficient quality of performed word processing.

2 Comparison of Morphological Parsers for Russian

We consider the most popular morphological processors that are freely available (so they can be tested) and are also frequently used in research projects,

¹ <http://aot.ru/docs/rusmorph.html>.

² <https://tech.yandex.ru/mystem/doc/>.

³ <http://corpus.leeds.ac.uk/mocky/>.

⁴ <http://pymorphy2.readthedocs.io/en/latest/index.html>.

⁵ <https://github.com/alesapin/XMorphy>.

⁶ <http://opencorpora.org>.

namely: AOT, Mystem, TreeTagger, Pymorphy2. Meanwhile, they demonstrate existing variety in functionality and approaches to build morphology models. The standard functionality of morphologic processors encompasses:

- lemmatization or/and stemming of a given word form;
- tagging its morphological features, first of all, POS (part of speech) and also gender, case, person, time, etc.;
- sufficient coverage of lexicon, which depends on used morphology model; for dictionary models it involves an ability to classify unknown (non-dictionary) words;
- generation of necessary word forms (or the whole word paradigm) for a given lemma.

We should note that in last two decades a trend appeared and settled to additionally provide parsers with properties that were earlier implemented by separate modules, namely:

- preliminary tokenization (and even sentence segmentation) of the text to be morphologically analyzed;
- morphological disambiguation of output parsing variants.

The reasons are obvious: traditional stages of text analysis, such as text segmentation, morphological analysis, syntax parsing correspond to language levels, which are internally interconnected. The results and quality of morphological analysis often strongly depend on text segmentation results: in Russian, typical examples are hyphen words, including specific terms, e.g. *α-редукция* (*α-reducing*), *интернет-новости* (*internet news*) – the hyphen is often omitted and should be restored. Specific writing forms of numerals, e.g., *3-й*, *32-ая* (*3rd, 32nd*), are also need special rules of processing, which are easier to implement as initial step of morphological analysis. As for morphological disambiguation, it facilitates subsequent syntactic analysis.

Besides above-mentioned pure linguistic properties of morphological processors, several more technological features are no less important for research projects and development of particular NLP applications. By technological features we mean:

- tools for modifying or/and extending morphological dictionary, as a rule, it means certain ability to connect a specific dictionary of your own;
- open source code, which makes it possible to integrate the source code of morphological parser into NLP programming project.

Both linguistic and technological properties are important to make an appropriate choice of tools for morphological processing in a particular application.

Comparing parsers AOT, Mystem, Tree Tagger, Pymorphy2 across the linguistic features, one can see that all of them perform lemmatization, full morphological tagging, and processing of non-dictionary words. Almost all compared parsers (except TreeTagger) are built on dictionary morphology models, while *TreeTagger* is built by training on tagged corpus [13]. At the same time, the dictionary-based parsers differ in accepted model of Russian morphology involving syntactic

classes of words (such as POS), and as a result, they have different systems of morphological tags and rules of lemmatization [8]. In particular, MyStem partially retains the canonical morphological paradigm inherited from Zaliznyak's grammar dictionary [16], and for word form *nponул* (*drunk*) it gives lemma with changed verb aspect *nponиeамъ* (*drink*) instead of expected lemma *nponумъ* (*drink away*), which is output by AOT and Pymorphy2.

The differences also concern processing of new (non-dictionary) words. In the parsers under comparison, prediction of lemma and morphological tags are based on various heuristics rules, so the results may essentially vary, for example, from four parsing variants for *Пикачу* (*Pikachu*) in Pymorphy2 and only one variant in MyStem.

Morphological disambiguation is important for Russian, since morphological homonymy is a hard problem for all higher flexional languages: in Russian texts, for almost each word form it is necessary to choose from 2-5 parsing variants generally differing in part-of-speech (POS), lemma and grammatical properties. Morphological disambiguation is absent in AOT parser, the other parsers implement various methods: non-contextual disambiguation in Pymorphy2 and more reliable statistical contextual disambiguation in MyStem and TreeTagger.

Generation of correct word forms is a more rare function of the parsers (it is not required in many NLP tasks), it is incorporated in AOT and Pymorphy2, while absent in MyStem and TreeTagger.

As for technological features, only two parsers, AOT and Pymorphy2 have an open source code, and only MyStem permits connection of a specific dictionary (by replacing the main dictionary, which is often not suitable).

Thus, the parsers under comparison vary in linguistic and technological features, and choice of a parser adequate for a particular NLP task may be difficult because of absence of necessary functionality. For development of our project based on lexico-syntactic patterns for building information extraction applications on the basis of surface syntactic analysis [4], we need an open source dictionary-based morphological processor with the main linguistic functions (lemmatization, morphological tagging, disambiguation), and also with stemming and word generation (in order to extract word phrases in correct grammatical form). Morphemic parsing of words are needed for our purposes as well, this makes it possible to recognize semantically close words (with the same root and some different affixes), such as *сахарный* and *сахаристый* (*sugar* and *sugary*), as well as words having different POS but indicating the same concepts, such as *компиляция* and *компилятор* (*compilation* and *compiler*).

Initially, we used processor AOT with open source code in our project. However, it is not supported now, its dictionary contains many obsolete words whereas does not include many new words, moreover, it does not provide morphological disambiguation. Among the other considered parsers, Pymorphy2 [7] has nearly sufficient functionality, but it provides simplest tokenization (it outputs only Russian words, the other tokens are skipped), it does not perform contextual disambiguation, and it is implemented in interpretive programming language Python, which complicates its integration into projects in other programming

languages. For these reasons, morphological processor CrossMorphy with the desired functionality has been built.

3 CrossMorphy: Key Decisions and Main Functions

Key decisions involve the choice of a computer model for Russian morphology and corresponding vocabulary data. Among the known models, the dictionary models based on large lists of possible word forms or word stems are traditionally used because of their linguistic quality (in particular, Zaliznjak's canonical model and dictionary [16] were implemented in almost all first known morphological analyzers including Russian version of Microsoft Word).

Initially, we made an attempt to build morphological processor based on the morphological model used in CrossLexica system [3], since it encompasses wide Russian lexicon significantly renewed at last decades. However, CrossLexica proposes too few morphological and lexical tags of words, so we decide to lean on vast and freely available OpenCorpora dictionary [2] with the detailed system of lexical and grammatical tags.

Thus, based on Open Corpora data, CrossMorphy's dictionary of word forms (~ 2 mln forms) was developed, taking the form of directed acyclic word graph (DAWG) [5], or acyclic finite state automaton. This effective data structure was proposed for storing word forms for highly flectional languages, and the same structure was applied in Pymorphy2 parser [7]. Therefore, CrossMorphy mainly inherits the system of lexical and morphological tags of OpenCorpora. Several rare grammatical cases were excluded (in particular, the second genitive), the traditional denotation of instrumental case was restored. Some modifications of OpenCorpora dictionary data were also made: several errors were fixed, analyses of single letters were excluded, lemma for personal pronouns was corrected, and links between adverbs and comparative adjectives were added, e.g., *дорого – дороже* (*expensive - more expensive*).

CrossMorphy performs both lemmatization and full morphological parsing, it is capable to find all interpretations of a given word forms. Stemming (that is splitting a given word form into pseudo flexion and pseudo stem and then outputting the latter) is incorporated into the processor as well (e.g., *носок*, *носками – нос*).

To estimate coverage of Russian lexicon, we have experimentally compared the rate of dictionary word forms processed by MyStem, Pymorphy, and CrossMorphy in vast text collection Librusec⁷, the results are 97.2%, 96.5%, and 96.6% correspondingly, which evidences the sufficient coverage. In comparison with MyStem, CrossMorphy proposes (on average) more parsing variants for homonymous word forms, in particular, for word *улыбающийся* (*smiling*) it gives 4 variants whereas MyStem has 2 variants.

CrossMorphy can generate both paradigm or particular word forms for a given lemma or word form. More precise, if input set of tags for a given word

⁷ <http://lib.rus.ec/>.

is incomplete, the processor produces all possible word forms. For example, for input word *шарами* and the given tag of grammatical number (single), CrossMorphy outputs the following forms: *шар*, *шара*, *шары*, *шаром*, *шаре*.

Functionality of CrossMorphy also includes no less useful auxiliary function of preliminary tokenization of texts and classifying tokens into words, numbers, punctuation, separators, and hieroglyphs. Each class of tokens has own additional tags, such as Cyrillic or Latin for words.

Handling of non-dictionary words and morphological disambiguation are also incorporated into CrossMorphy.

4 Processing of Non-dictionary and Hyphen Words

For handling new (non-dictionary) words and predicting their morphological features, CrossMorphy applies three general heuristic methods.

Prediction according word flexion (ending) is based on the well-known principle of analogy used in almost in all parsers for Russian with dictionary morphology. As a rule, the same word endings (1-5 last letters) correspond to the same syntactic class, so morphological tags (and lemma) of unknown word may be predicted by the final letters. The implementation of the principle varies in morphological processors, giving different numbers of resulting variants.

We propose the following prediction version with a reasonable number of answers. Statistics on all word endings (1 to 5 letters long) are collected for the dictionary, rare endings encountered less than 3 times are excluded, and the most frequent POS (part of speech) are determined for any particular ending. Then all the morphological interpretations for the endings with the determined POS are considered as the result.

Prediction according prefix is the second method, it involves cutting of possible prefix and then parsing the rest of the word form. Unlike AOT and Pymorphy2, we take into account only known prefixes (the built-in list of 207 prefixes compiled in open Russian Wiki-dictionary⁸ is used) – this makes it possible to avoid errors in prediction of some words (e.g., for word *вейнэр*).

For handling unknown hyphen words several rules are employed, accounting for cases with several hyphen (e.g., *фолк-панк-рок* – *folk-punk-rock*), words with digital and Latin letters constituents (*Рубин-5*, *S-выражение* – *Ruby-5*, *S-expression*), words with a single inclined constituent (*веб-инструктор* – *web instructor*), and with both inclined constituents (*человек-гора* – *man-mountain*).

It is important that three described methods are applied independently, and as a result, parsing of some words (such as *автомаша*) is successive whereas in Pymorphy2 it fails.

To estimate processing of non-dictionary words, we used tagged corpus of NCRL (National Corpus of Russian Language)⁹ with ~ 1 mln word forms. Table 1 presents comparative data counted for three parsers: the total number of

⁸ <https://ru.wiktionary.org/wiki/>.

⁹ <http://ruscorpora.ru/>.

encountered non-dictionary tokens, percentages of tokens with correct resulted lemma, POS, and full tag parses accordingly (all the parsers performed morphological disambiguation). One can see that CrossMorphy wins in POS accuracy, exceeds Pymorphy2’s scores for full tags, but loses to Mystem in lemma and full set of tags.

Table 1. Accuracy of parsing non-dictionary words

Processor	Total #	Lemma (%)	POS (%)	Full tags (%)
Mystem	11478	66.20	72.58	56.51
Pymorphy	15024	60.43	67.15	35.71
CrossMorphy	15030	59.68	85.60	41.13

5 Morphological Disambiguation

To now, the problem of POS classification for wordforms is well investigated for many languages, and disambiguation accuracy is near 98%. One of the first work for Russian [11] proposed the statistical method with the accuracy 97.42%, while MorphoRuEval-2010 evaluation [9] reported 94-95% obtained by rule-based methods.

For Russian, the challenging task is full morphological disambiguation, i.e. assignment of lemma and all meaningful grammatical tags (POS, case, gender, person, etc.) to word token. In the recent work [10] CRF (conditional random field) statistical method for morphological disambiguation was investigated and resulted in the accuracy up to 94, 95%.

We should note that all indicated evaluation rates are relative, since they depend on several factors including not only the applied method, but also the set of used morphological tags and the size of text corpora for training and testing. In all the works mentioned above, these factors differ, and at the same time all of them use reduced tag sets, as well as relatively small test corpora. In overall, evaluation and comparison of disambiguation method is complicated by the fact that there is neither standard of Russian morphology tagging, nor gold standard corpora for evaluation. Besides, the real problem is some incompatible tags used in morphological parsers.

In CrossMorphy two methods of statistical morphological disambiguation are implemented, contextual and non-contextual. The latter ranks parsing variants for a processed word form, according to frequency statistics of all parsing variants for the corresponding lemma. The statistics are gathered on the tagged corpus of NCRL (National Corpus of Russian Language)¹⁰. The possibility of a particular parsing variant is calculated according the formula

$$P(t|w) = \frac{Fr(w, t)}{Fr(w)}$$

¹⁰ <http://ruscorpora.ru/>.

where w is a word form, t is a set of morphological tags, $Fr(w)$ and $Fr(w, t)$ are frequencies of w and w with its tags t in the corpus. Similar to the other parsers, CrossMorphy outputs the calculated probabilities of homonymous variants.

MyStem and Pymorphy2 use another methods to compute non-contextual scores of parsing variants, but it makes no sense to compare them, since they only rank the parsing variant, and the resulted ranks are similar.

In CrossMorphy, the non-contextual method is considered as auxiliary for contextual disambiguation. For the latter, CRF++ method is applied, so far as it presents results close to the state of the art for POS tagging for flexional languages [10, 12]. Specifically, we use Limited-memory BFGS version of CRF. Since our classification task involves too many features (POS, lemma and all Russian obligatory grammatical tags), four CRF classifiers are sequentially applied.

First, the POS classifier is used, among accounted features are the token being processed and possible POS variants in the form of binary vector. The next CRF classifier is responsible for gender recognition, and accounted features are lemma, POS determined by the previous classifier, and also possible variants of gender (masculine, feminine, neutral). In similar way, subsequent CRF classifiers for number and case work. After all the classifying procedure, rare homonymous variants could still remain (for example, concerning animacy), in this case the non-contextual disambiguation is applied to choose an adequate variant.

An example of disambiguation for Russian word form *мыла* (*washed* or *soap?*) is shown in Fig. 1.

мыла	мылю	NOUN,inan,neut,nomn,plur
мыла	мылю	NOUN,gent,inan,neut,sing
мыла	мылю	NOUN,accs,inan,neut,plur
мыла	мыть	VERB,femn,impf,indc,past,sing,tran

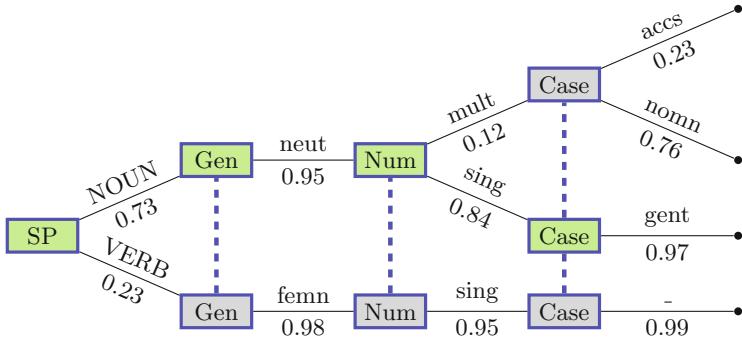


Fig. 1. Disambiguation of word form *мыла*

To estimate CrossMorphy's disambiguation model, we have performed two experiments with training the model on tagged data and its cross-validation (10%). We first used the corpus of NCRL with ~ 1 mln word forms, and then Syntagrus and GICRL (General Internet Corpus of Russian Language)

tagged data with more than 2 mln tokens. The latter corpora were obtained within MorphoRuEval-2017 competition [15] for comparing various disambiguation methods on the basis of large tagged Russian text corpora and the system of Universal Dependency tags¹¹. In both experiments we had to convert morphological tags: NCRL tags into CrossMorphy's tags, and CrossMorphy's tags into UD tags (in the latter case we had to resolve some mismatches of the tag sets, concerning, in particular, restoring the difference between the comparative adjectives and comparative adverbs).

Accuracy rates achieved in the experiments after each step of the overall CRF classification procedure are presented in Table 2 (the last column also indicates full tag disambiguation). The rates of POS classification are better than in [9, 11], but the final full tag rates (90-93%) are slightly less than in [12, 15]. The best result achieved in the closed track of MorphoRuEval-2017 [15] is 93.39 (while the open track gives 97.11 due to training on large corpora and using neural net models). Thus, a more tricky procedure should be further developed for CrossMorphy. Our experiments also showed that another sequences of classifying gender, number and case do not improve final accuracy of diambiguation.

Table 2. Accuracy of sequentially applied CRF classifies (%)

Corpora / CRF	POS	Gender	Number	Case
NCRL	97.94	97.03	96.61	93.42
Syntagrus+GICRL	98.12	96.32	94.23	90.53

Accuracy of disambiguation showed by CrossMorphy indirectly evidences the quality of its dictionary and procedures for handling non-dictionary words. What is important for us, that CrossMorphy demonstrates about similar behavior on various testing corpora, containing news, fiction, and texts from internet social networks. Taking into account that conversion of morphological tags, which is needed for training, may lead to inevitable loss of significant information, we think that there is a reserve to improve overall quality of CrossMorphy parsing, in particular, disambiguation accuracy.

6 Morphemic Parsing

Additional functionality supported by CrossMorphy is automatic morphemic parsing (segmentation), that is dividing words into their morphs (root and affixes), e.g. *нод-ко^вер-и-и^й, е-бр^ас-и^е-ам^ь-с^я, ии-д^ык^у-и-и*. Clearly, it is reasonable to store morphemic structure of words in the dictionary, but there exist significant problems. First, there are no full dictionaries with morphemic segmentation of words (and many words of OpenCorpora and CrossMorphy are

¹¹ <http://universaldependencies.org/u/overview/morphology.html>.

absent in the known dictionaries). Second, there is no agreement between linguists about rules of morphemic segmentation for Russian words (apart from another languages with rich morphologies, there are many affixes of various types and behavior in Russian). And finally, the task cannot be automatically solved with high accuracy because of similarity of morphs.

Unlike the works [1, 6, 14], for automatic morphemic parsing we use supervised machine learning, specifically, CRF method. Morphemic segmentation is considered as classification of letters by recognizing their morphemic classes (Prefix, Root, Suffix, Ending). As accounted features we take the letter itself, is it a vowel, lengths of the word and its stem, POS of the word, its morphological tags, and also Harris's features [6] (local maximums of letter frequencies counted for various positions within words). An example of resulted classification is showed in Fig. 2.



Fig. 2. Morphemic parsing of word *индукция*

Morphemic classification models were obtained by training on two tagged data taken correspondingly from CrossLexica system [3] (23426 parsed words) and Russian Wiki dictionary¹² (94485 parsed words). We could not combine these two data sets, since many words presented in both sets have different morphemic segmentation. Thus, we separately built two classifiers, and their accuracy was evaluated both on fragments of the own and alien corpora – the results are presented in Table 3.

Table 3. Accuracy of morphemic classifiers

Data		Precision				
Training	Validation	Whole word	Prefix	Root	Suffix	Ending
CrossLexica	CrossLexica	74.2	86.13	75.10	77.13	97.95
CrossLexica	Wiki	35.91	66.14	56.75	38.57	57.35
Wiki	CrossLexica	46.64	78.11	66.38	50.43	70.20
Wiki	Wiki	65.87	70.92	65.47	71.84	98.31

One can see that cross validation on the alien corpus gives a significant loss of accuracy. CrossLexica's data yields the best scores for the most morphemic classes (unlike Wiki dictionary, the data were created by a single human expert, so are more homogeneous). For this reason, corresponding classifier was incorporated into our processor. The accuracy of the incorporated classifier (74,2%) is better than the best result 70% obtained for Turkish in [14].

¹² <https://ru.wiktionary.org/wiki/>.

7 Conclusions and Future Work

In this paper we have compared functional properties of several popular freely available morphological processors for Russian texts, thus explaining the reasons to develop yet another processor for Russian. The developed open source morphologic processor CrossMorphy has the distinguishing combination of properties that meets our requirements. Evaluation of its functionality has showed sufficiently accurate processing of Russian texts. Across main functions, our processor is competitive with known freely available parsers, and at the same time its functionality is extended by morphemic segmentation.

On the way towards a high-quality morphological processor, the further improvements of CrossMorphy are needed:

- more exhaustive testing and providing convenient documentation;
- providing tools for connecting user dictionaries;
- incorporating additional rules for classifying non-dictionary words based on information about their morphemic segmentation;
- elaborating a more accurate model of morphological disambiguation;
- providing linguistically correct convertors between different systems of Russian morphological tags; creation of suitable universal system of morphological tags for Russian is a more challenging task.

Acknowledgements. We would like to thank the reviewers of our paper for their helpful comments.

References

1. Bernhard, D.: Simple morpheme labelling in unsupervised morpheme analysis. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 873–880. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-85760-0_112
2. Bocharov, V., Bichineva, S., Granovsky, D., Ostapuk, N., Stepanova, M.: Quality assurance tools in the opencorpora project. In: Computational Linguistics and Intelligent Technologies: Papers from the Annual International Conference “Dialogue” (2011)
3. Bolshakov, I.A.: CrossLexica, the universe of links between Russian words. In: Busyness Informatica, No. 3 (2013)
4. Bolshakova, E., Efremova, N., Noskov, A.: LSPL-patterns as a tool for information extraction from natural language texts. In: New Trends in Classification and Data Mining. Markov, K., et al. (eds.) ITHEA, Sofia, pp. 110–118 (2010)
5. Daciuk, J., Mihov, S., Watson, B., Watson, R.: Incremental construction of minimal acyclic finite state automata. Comput. Linguist. **26**(1), 3–16 (2000)
6. Harris, Z.S.: Morpheme boundaries within words: report on a computer test. In: Transformations and Discourse Analysis Papers, vol. 73, pp. 68–77 (1970)
7. Korobov, M.: Morphological analyzer and generator for Russian and Ukrainian languages. In: Khachay, M.Y., Konstantinova, N., Panchenko, A., Ignatov, D.I., Labunets, V.G. (eds.) AIST 2015. CCIS, vol. 542, pp. 320–332. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26123-2_31

8. Kuzmenko, E.: Morphological analysis for Russian: integration and comparison of taggers. In: Ignatov, D.I., Khachay, M.Y., Labunets, V.G., Loukachevitch, N., Nikolenko, S.I., Panchenko, A., Savchenko, A.V., Vorontsov, K. (eds.) AIST 2016. CCIS, vol. 661, pp. 162–171. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-52920-2_16
9. Ljashevskaya, O., Astaf'eva, I., Bonch-Osmolovskaja, A., Garejshina, A., Grishina, J., D'jachkov, V., Ionov, M., Koroleva, A., Kudrinskij, M., Litjagina, A., Luchina, E., Sidorova, E., Toldova, S.: NLP evaluation: Russian morphological parsers. In: Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference “Dialogue”, pp. 318–326 (2010)
10. Muzychka, S.A., Romanenko, A.A., Piontkovskaja, I.I.: Conditional random field for morphological disambiguation in Russian. In: Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference “Dialogue”, pp. 456–465 (2014)
11. Segalovich, I.A.: Fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. In: MLMTA, pp. 273–280 (2003)
12. Shen, Q., Clothiaux, D., Tagtow, E., Littell, P., Dyer, C.: The role of context in neural morphological disambiguation. In: COLING 2016, 26th International Conference on Computational Linguistics. Proceedings of the Conference: Technical Papers, Osaka, Japan. ACL, pp. 181–191 (2016)
13. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the International Conference on New Methods in Language Processing, pp. 44–49 (1994)
14. Smit, P., Virpioja, S., Gronroos, S., Kurimo, M.: Morfessor 2.0: toolkit for statistical morphological segmentation. In: Proceedings of the Demonstrations at the Conference of the European Chapter of the ACL, pp. 21–24 (2014)
15. Sorokin, A., Shavrina, T., Lyashevskaya, O., Bocharov, V., Alexeeva, S., Droganova, K., Fenogenova, A.: MorphoRuEval-2017: an evaluation track for the automatic morphological analysis methods for Russian. In: Computational Linguistics and Intellectual Technologies. Proceedings of International Conference Dialogue 2017, Moscow (2017)
16. Zaliznjak, A.A.: Grammatical Dictionary of Russian: Inflection. Russkij Jazyk Publisher, Moscow (1977)

SyntaxNet Errors from the Linguistic Point of View

Oleg Durandin^{1,2} , Alexey Malafeev¹ ,
and Nikolai Zolotykh^{1,2} 

¹ National Research University Higher School of Economics,
Nizhny Novgorod, Russia

{odurandin, aumalafeev, nzolotykh}@hse.ru

² Lobachevsky State University of Nizhny Novgorod, Nizhny Novgorod, Russia

Abstract. The paper deals with Google’s universal parser SyntaxNet. The system was used to analyze the Universal Dependencies linguistic corpora. We conducted an error analysis of the output of the parser to reveal to what extent the error types are connected with or preconditioned by the language types. In particular, we carried out several experiments, clustering the languages based on the frequency of different errors made by SyntaxNet, and studied the similarity of the resulting clustering with the traditional typology of languages. Three types of errors were separately considered: part-of-speech tagging, dependency labeling, and attachment errors. We show that there is indeed a correlation between error frequencies and language types, which might indicate that to further improve the performance of a universal parser, one needs to take into account language-specific morphological and syntactic structures.

Keywords: Natural language processing · Syntax parsing · SyntaxNet
Error analysis · Linguistic typology

1 Introduction

It is well-known that sentence structure in natural language is most commonly described in one of the two ways: either by recursively breaking it up into phrases (constituents) or by drawing links (dependencies) between individual words. The differences between constituency and dependency grammars are thoroughly discussed in (Matthews 1981).

In natural language processing, parsing has been an important task for the past decades. With the development of the Penn Treebank (Marcus et al. 1993), it became possible to train models on a sufficiently large collection of sentences in English, annotated for syntactic structure. Progress in data-driven parsing for English was gradually made in some relatively early works, notably (Eisner 1996; Collins 1997; Charniak 2000; Klein and Manning 2003; Collins 2003). 2006 saw a transition to multilingual parsing, which was largely due to the availability of new treebanks for other languages and competitions such as the CoNLL-X shared task on multilingual dependency parsing (Buchholz and Marsi 2006).

While earlier parsing systems mainly focused on constituency parsing, starting from about 2005 there was a continuous shift towards dependency parsing (Nivre 2005). According to McDonald et al. (2006, p. 216): “[t]his interest has generally come about due to the computationally efficient and flexible nature of dependency graphs and their ability to easily model non-projectivity in free-word order languages”. MaltParser, a language-independent system, was built and successfully used for data-driven dependency parsing of multiple languages (Nivre et al. 2006).

However, newer collections of multilingual treebanks with homogeneous syntactic dependency annotation, such as the one described in (McDonald et al. 2013), made it possible to achieve even higher accuracy in parsing. Ultimately, the corpora at the Universal Dependencies project (Nivre et al. 2016) became ample material for pushing the envelope even further. Very recently, Google’s SyntaxNet (Petrov 2016), an open-source neural network framework based on the models described in (Andor et al. 2016), achieved groundbreaking results (over 94% accuracy for English).

In this paper, we conduct a language-focused error analysis of SyntaxNet output. To the best of our knowledge, such an analysis has not been previously carried out. Furthermore, similar analyses based on the output of other multilingual parsing systems have not been described in literature either. That is, error analyses conducted by other researchers, such as (McDonald et al. 2006) or (McDonald and Nivre 2007), primarily focused on parsing methods and models, as well as system performance. In contrast, our analysis is aimed at clarifying connections between different types of parsing errors and language types.

2 Experiment Setting

Using SyntaxNet and pre-trained models¹ for the system, we parsed the test set from the Universal Dependencies project (version 1.6, 46 datasets for 34 languages). Standard evaluation measures were used: Part-of-Speech (PoS) accuracy, as well as labeled and unlabeled attachment score (LAS and UAS). Table 1 summarizes the results attained (in bold are the corpora further used in clustering, see the explanation below).

Having parsed the test corpora, we processed the .conll files that contained the resulting PoS and dependency annotations. Since the evaluation measures used (PoS tagging accuracy, LAS and UAS) do not provide sufficient information about the quality of syntax and morphology analysis, we used the following error classes further subdivided into numerous specific error types:

- 1. Part-of-speech tagging errors** (229 types). These have the form: *actual PoS tag* \Rightarrow *hypothetic PoS tag*, e.g. *verb* \Rightarrow *adj* means that a verb was incorrectly classified as an adjective.
- 2. Dependency labeling errors** (945 types). In these, the head and dependent words are correctly identified (a correct unlabeled dependency relation), but the relation type is labelled incorrectly. For these errors, the following notation is used: *actual*

¹ Made available by Google: <https://github.com/tensorflow/models/blob/master/syntaxnet/universal.md>.

Table 1. Languages analyzed with SyntaxNet

Language	Tokens	PoS	UAS	LAS	Source types
Czech	173918	0.9812	0.8947	0.8593	News
Russian [SyntagRus]	108100	0.9288	0.9131	0.8301	News, Nonfiction, Fiction
Catalan	59503	0.9806	0.9047	0.8764	News
Spanish [AnCora]	53594	0.9828	0.8926	0.8650	News
Hindi	35430	0.9645	0.9304	0.8932	News
Norwegian	29966	0.9743	0.8859	0.8620	News, Blogs, Nonfiction
Galician	29746	0.9681	0.8448	0.8135	Medical, Legal, Nonfiction, News
Portuguese [BR]	29438	0.9707	0.8791	0.8544	News, Blogs
Arabic	28268	0.9565	0.8149	0.7582	News
English	25096	0.9019	0.8480	0.8040	Blogs, Social, Reviews
Basque	24374	0.9488	0.7800	0.7336	News, Fiction
Estonian	23670	0.9592	0.8310	0.7883	Fiction, News, Science
Swedish	20377	0.9627	0.8384	0.8028	News, Nonfiction
Finnish [FTB]	16286	0.9350	0.8497	0.8048	Grammar examples
German	16268	0.9182	0.7973	0.7135	News, Reviews, Wiki
Persian	16024	0.9622	0.8440	0.8025	News, Fiction, Medical, Legal, Social, Spoken, Nonfiction
Bulgarian	15734	0.9771	0.8935	0.8506	News, Legal, Fiction, Misc
Latin [PROIEL]	14906	0.9650	0.7760	0.7098	Bible, Nonfiction
Slovenian	14063	0.9622	0.8771	0.8460	News, Nonfiction, Fiction
Hebrew	12125	0.9504	0.8461	0.7871	News
Chinese	12012	0.9132	0.7671	0.7124	Wiki
Italian	10952	0.9728	0.8980	0.8689	Legal, News, Wiki
Czech [CAC]	10862	0.9811	0.8728	0.8344	News, Nonfiction, Legal, Reviews, Medical
Russian	9573	0.9065	0.8177	0.7681	Wiki
Finnish	9140	0.9478	0.8365	0.7960	News, Wiki, Blog, Legal, Fiction, Grammar-examples
Turkish	8616	0.9365	0.8195	0.7134	News, Nonfiction
English [Lines]	8481	0.9534	0.8150	0.7737	Fiction, Nonfiction, Spoken
Swedish [Lines]	8228	0.9600	0.8138	0.7721	Fiction, Nonfiction, Spoken
Spanish	7953	0.9527	0.8506	0.8153	Blogs, News, Reviews, Wiki
Polish	7185	0.9505	0.8830	0.8271	Fiction, Nonfiction, News
French	7020	0.9645	0.8466	0.8105	Blogs, News, Reviews, Wiki

(continued)

Table 1. (*continued*)

Language	Tokens	PoS	UAS	LAS	Source types
Latin[ITTB]	6548	0.9798	0.8422	0.8117	Nonfiction
Croatian	6306	0.9420	0.7942	0.7293	News, Web, Wiki
Portuguese	6294	0.9635	0.8419	0.8004	News
Danish	5884	0.9528	0.7984	0.7634	News, Spoken, Fiction, Nonfiction
Dutch	5843	0.8989	0.7770	0.7121	News
Greek	5668	0.8155	0.8368	0.7999	News, Wiki, Spoken
Gothic	5158	0.9558	0.7933	0.7169	Bible
Latin	4832	0.8804	0.5600	0.4580	Fiction, Nonfiction, Bible
Dutch [LassySmall]	4562	0.9562	0.8163	0.7808	Wiki
Hungarian	4235	0.9400	0.7875	0.7183	News
Czech [CLTT]	4105	0.9579	0.7734	0.7340	Legal
Latvian	4075	0.8083	0.5890	0.5158	News
Irish	3821	0.9134	0.7451	0.6629	News, Fiction, Web, Legal, Media
Slovenian [SST]	2951	0.9000	0.6506	0.5696	Spoken
Tamil	1989	0.7929	0.6445	0.5535	News

relation \Rightarrow *hypothetic relation*. For example, *nsubj* \Rightarrow *obj* means that a nominal subject dependency relation was erroneously classified as an object relation.

3. **Attachment errors** (2036 types). In these error cases, the link between some two words was identified incorrectly. Let us first introduce the notation $PoS1 \rightarrow PoS2$, meaning a head being a $PoS1$ (e.g. a noun) is connected with a dependent being a $PoS2$ (e.g. an adjective). Thus, the error itself can be denoted as follows: $PoS1 PoS2 \Rightarrow PoS3 \rightarrow PoS4$. Importantly, the dependent is always the same in both the actual (to the left of \Rightarrow) and hypothetic (to the right of \Rightarrow) attachments.

For example, consider the fragment *her charming smile*. The correct attachment of *charming* is to *smile*:



Let us suppose that the parser yields the following (incorrect) attachment:



In this case, the error can be denoted as *noun* → *adj* ⇒ *pron* → *adj*, which means that a dependent adjective (*adj*) is incorrectly attached to a pronoun (*pron*) as the head, while the correct attachment is to a noun.

We counted the frequencies of each error subtype within each of the three general error classes (PoS tagging, dependency labeling and attachment), thus building three confusion matrices for each corpus. For more uniformity, we had excluded very small corpora (of less than 8000 tokens), as there is not enough data for reliable error analysis. We also excluded those corpora that are large in comparison with others (of more than 100,000 tokens), as preliminary experiments showed that these are very unbalanced, which causes a lot of noise in clustering. Also, we excluded the corpora for the same languages (Swedish[Lines], English[Lines], Portuguese, Latin[ITTB]). After this, 23 corpora/languages remained, listed here alphabetically: Arabic, Basque, Bulgarian, Catalan, Chinese, Czech[CAC], English, Estonian, Finnish[FTB], Galician, German, Hebrew, Hindi, Italian, Latin[PROIEL], Norwegian, Persian, Portuguese[BR], Russian, Slovenian, Spanish[AnCora], Swedish, Turkish.

3 Experimental Study

To better understand the nature of SyntaxNet errors and whether these are related to language types, we performed hierarchical clustering of languages using error frequencies as parameters. After this, an attempt was made to interpret the results from a linguistic (typology) point of view.

For hierarchical clustering, we used the grouping algorithm based on the Ward’s method (Ward 1963) and the Pearson correlation metric. The advantage of correlation-based metrics is that they are unit independent; also, they are more relevant for the given task, since we are interested in how strong the interrelation between languages is and how this affects the parser errors.

3.1 Experiment 1: Analysis of PoS Tagging Errors

First, hierarchical clustering was performed based on PoS tagging errors. During this phase of the analysis, 23 languages with 229 numerical attributes were processed. The attributes were simply frequencies of each PoS tagging error type. Figure 1 below shows a dendrogram of hierarchical clustering (with top 11 clusters highlighted).

Note that the position of each split on the x-axis depends on how similar the two languages/groups are (the closer to the right, the more similar). It can be seen from Fig. 1 that one of the clusters is formed by Galician, Catalan and Spanish, which can be easily interpreted, as these languages are highly related. Also, the Estonian, Finnish and Basque languages are found in one cluster; these languages are of the agglutinative morphological type with some elements of inflection. The Persian and Turkish languages also form a cluster, and from the linguistic point of view, these languages have a common property: their agglutinative morphological type with strict suffixation. Slovenian and Bulgarian both belong to the South Slavic language group, while the Czech language is associated with the West Slavic language group, and quite

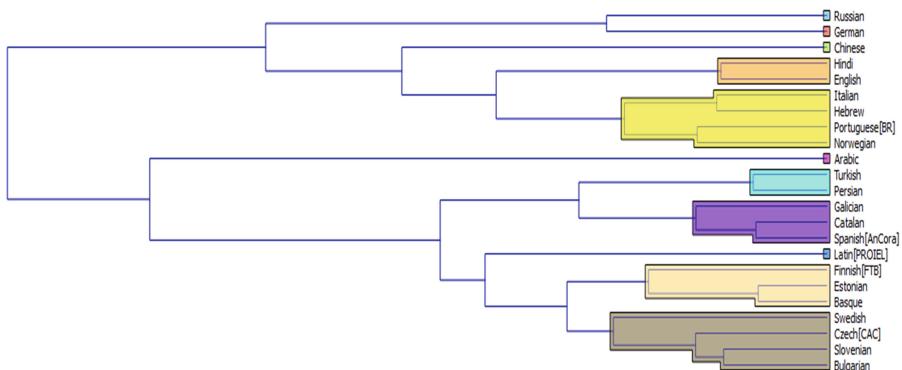


Fig. 1. Hierarchical clustering based on PoS tagging errors

consistently with this fact, these languages are in the same cluster. However, Swedish (but not Russian) is also found in this cluster, although the “distance” from Swedish to the other languages in this cluster is quite large.

It is quite difficult to give a linguistic explanation to the grouping of English with Hindi: it is not clear why SyntaxNet’s PoS tagger makes similar errors while processing these very different languages. Another cluster that is hard to interpret includes the Hebrew, Italian, Portuguese, and Norwegian languages. While Hebrew is of the Afro-Asiatic (specifically, Semitic) language family, the other three belong to the Indo-European family.

We also ensured that the resulting clustering was not simply due to the distribution of parts of speech in each of the languages. To this end, we used PoS frequencies as parameters to perform another hierarchical clustering, using the same correlation metric and the same grouping algorithm. As can be seen from Fig. 2 below, the new clustering is quite different, although there are certain similarities to the previously shown clustering (as in the case of Catalan and Spanish, for example).

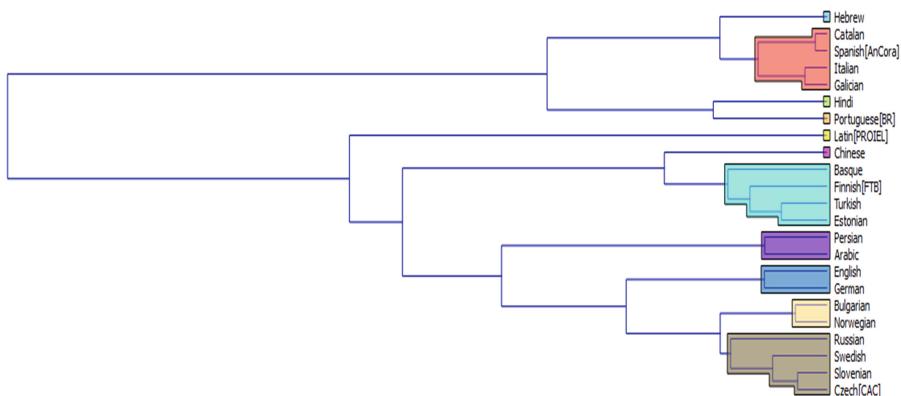


Fig. 2. Hierarchical clustering based on PoS distribution in the test corpora

3.2 Experiment 2: Analysis of Dependency Labeling Errors

SyntaxNet performs first PoS tagging, then, based on that, parsing. Therefore, some of the system’s syntax errors result from PoS tagging errors. To negate the influence of PoS tagging accuracy on parsing accuracy, when processing the test corpora we used the “ground truth” PoS tags given rather than the hypothetic PoS labels from SyntaxNet. Unfortunately, it is not fully clear from the documentation available on <https://github.com/tensorflow/models/blob/master/syntaxnet/g3doc/universal.md> whether Google’s parsing models themselves were trained on gold standard PoS tags or on the ones coming out of the tagging models. Figure 3 shows Pearson correlation-based hierarchical clustering for dependency labeling errors (23 languages, 945 numeric attributes).

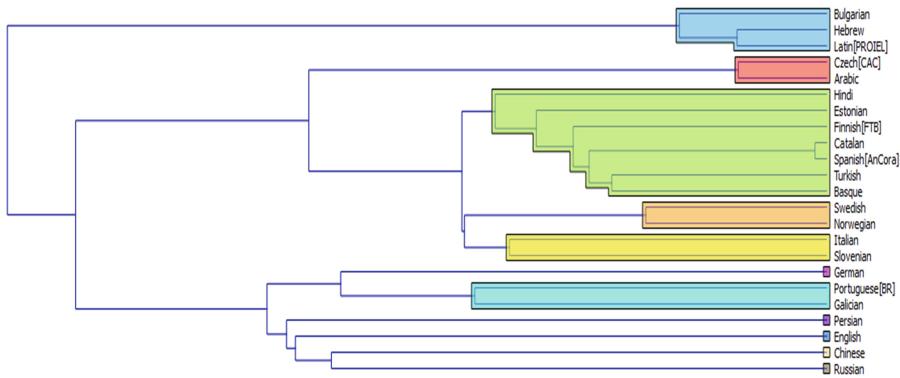


Fig. 3. Hierarchical clustering based on dependency labeling errors

In this case, the clustering is not as easily interpretable from the linguistic typology viewpoint as in the previous experiment. We will assume that syntax trees also depend on the type of text found in the corpus. Indeed, a great body of linguistic research suggests considerable syntactic variation that is register-dependent: see, for example, (Ferguson 1983; Haegeman 1990; Ferrara et al. 1991). Therefore, we will consider a subset of languages whose corpora consist of news articles and non-fiction, because these are quite similar in register. We thus remove four languages/corpora: Chinese, English, Finnish, and Russian. The dendrogram resulting from hierarchical clustering using the same parameters as before, but a smaller subset of languages, is presented in Fig. 4. Again, here and further on we use the frequency data on parsing errors made while processing corpora with “ground truth” PoS tags, to negate the possibility of SyntaxNet automatic morphological annotation affecting the results of parsing.

This clustering still poses some difficulty for linguistic interpretation. On the one hand, some groupings contain closely related languages of similar structure, namely Catalan and Spanish, as well as Swedish and Norwegian. Some other groupings include languages that are not related, yet share some important linguistic features. In

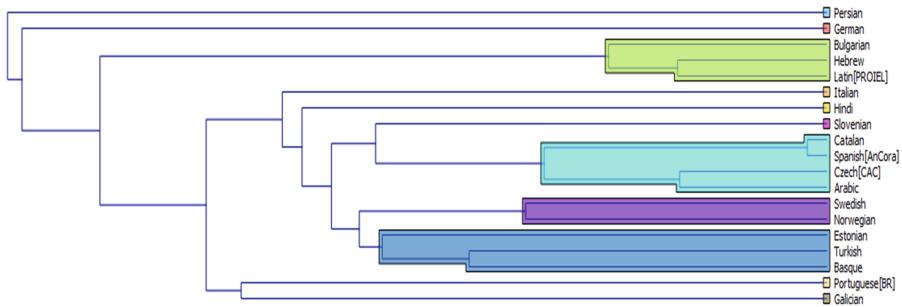


Fig. 4. Hierarchical clustering based on dependency labeling errors, of a subset of languages whose corpora consist of news and non-fiction

particular, Estonian, Turkish and Basque are not related, but they are characterized by morphological richness. Turkish and Basque also share a common word order, SOV (subject, object, verb), while in Estonian, it is SVO (subject, verb, object). Interestingly, as shown in the dendrogram, Turkish and Basque are grouped more closely together, while Estonian is slightly further from them.

Hebrew and Latin are not related; however, in both these languages, inflection plays a decisive role in the formation of verbs and nouns. As for Bulgarian, from the linguistic perspective it should be grouped with the other Slavic languages, namely Slovenian and Czech, rather than in the same cluster with Hebrew and Latin. Yet the grouping here does not seem to closely follow linguistic typology, with Czech being clustered with Arabic.

Overall, it can be said that with dependency labeling errors there might be more noise than with PoS tagging errors, so the clustering is not as consistent with language affinity and shared structural properties of the languages. Thus, due to the nature of dependency labeling errors, when two words are correctly linked, but the label for dependency type is incorrect, it might be the case that this class of errors is more language-specific. However, there is still some correlation between structural commonalities of the languages and their hierarchical clustering based on dependency labeling errors.

Like in Sect. 3.1, where we considered PoS distribution in the languages, we also calculated the dependency frequencies in each of the corpora used and performed a clustering based on this frequency data. The resulting dendrogram is shown in Fig. 5 below. In contrast to Fig. 4, this clustering is much more consistent with language affinity, which is not surprising: languages similar in syntactic structure are expected to have similar distributions of dependency types.

3.3 Experiment 3: Analysis of Attachment Errors

In the third experiment, an analysis of attachment errors was carried out. For this class of errors, 2036 attributes were obtained, and clustering was performed using the same parameters as in the previous two experiments. Like in Sect. 3.2, “ground truth” PoS

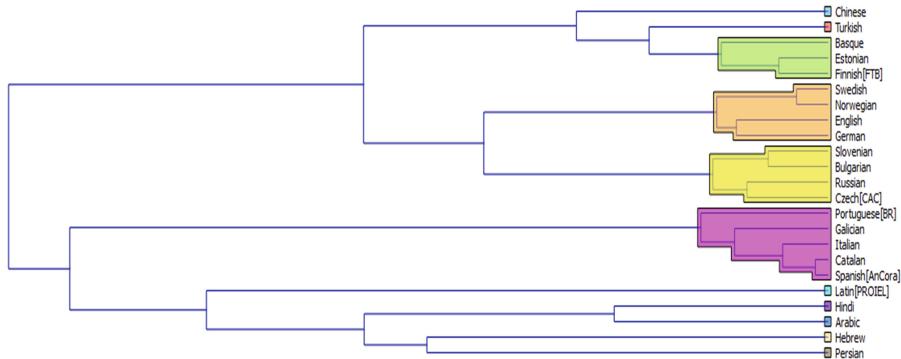


Fig. 5. Hierarchical clustering based on dependency type frequencies in the test corpora

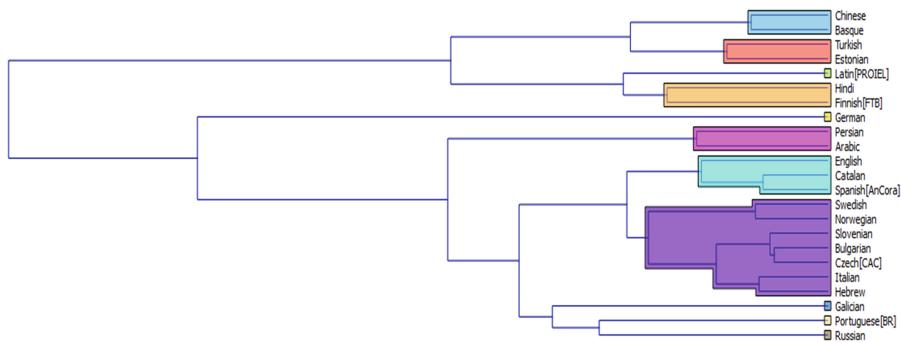


Fig. 6. Hierarchical clustering based on attachment errors

tags were used to negate the influence of PoS tagging errors. The resulting dendrogram is presented in Fig. 6.

This dendrogram has some groupings that are difficult to interpret from the linguistic viewpoint, such as Chinese and Basque, or Hindi and Finnish. Like in the previous experiment, we will consider only the corpora comprised by news and non-fiction. The resulting clustering is shown in Fig. 7.

As can be seen, this dendrogram is more interpretable from the point of view of linguistic typology. Turkish and Estonian are agglutinative languages, and are grouped together here. Despite the fact that Persian and Arabic belong to different language families, they have influenced each other for a long time, although structurally these languages are quite different. The groupings of Catalan and Spanish, Swedish and Norwegian, as well as Slovenian, Bulgarian and Czech, all have valid linguistic grounds, since each of these three groupings contains closely related languages. In contrast, the clustering of Italian and Hebrew is still difficult to explain.

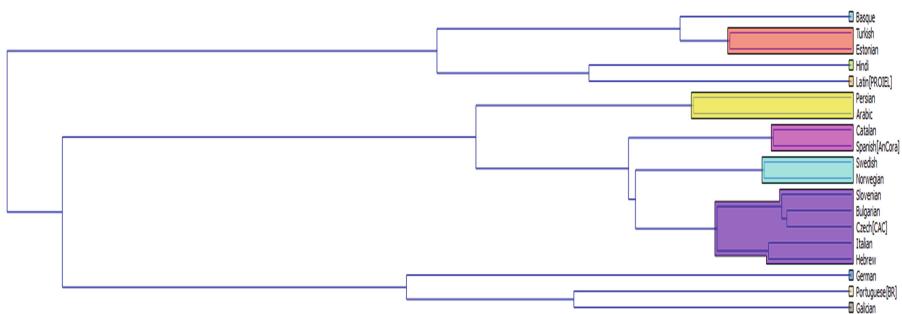


Fig. 7. Hierarchical clustering based on attachment errors for news and non-fiction sources

3.4 Experiment 4: Analysis of Combined Syntax Errors

The next step was to perform hierarchical clustering based on combined syntax errors, i.e. both dependency labeling and attachment errors (totaling 2,981 attributes). The results are presented in Fig. 8.

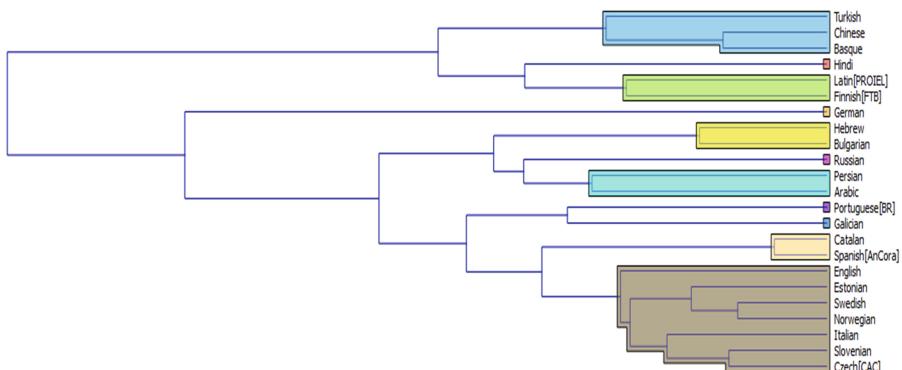


Fig. 8. Hierarchical clustering based on combined syntax errors

Like in the two previous experiments, we will also assume that the register of the source texts comprising the corpora has a significant influence on parsing errors. Therefore, we will consider only the corpora formed by news and non-fiction (Fig. 9).

It is difficult to explain the grouping of Hindi and Estonian. In Estonian, the word order is SVO, while in Hindi it is SOV, although less rigid. There are two noun cases in Hindi, and nouns are also inflected for number and gender, while in Estonian there are 14 cases and no grammatical gender.

The Turkish and Basque cluster might be easier to interpret. Despite not being related, these languages have important common morphological features: an abundance of suffixes, as well as a similar word order, SOV.

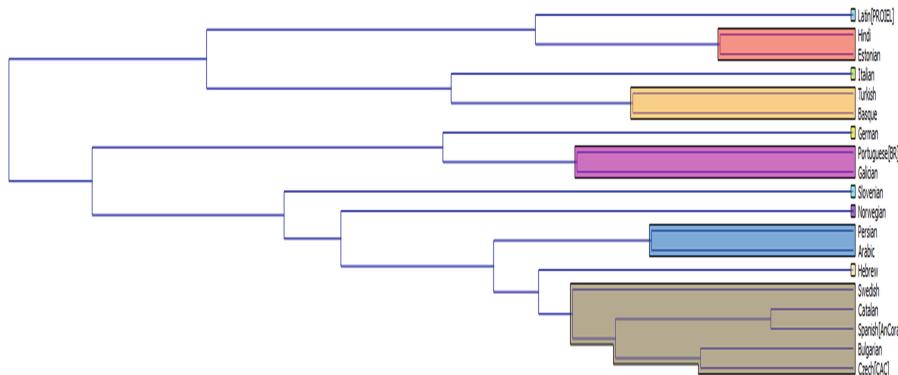


Fig. 9. Hierarchical clustering based on combined syntax errors for news and non-fiction sources

Portuguese and Galician are closely related. The pairwise grouping of Spanish and Catalan, as well as Bulgarian and Czech, is also consistent with language affinity. Together with the Swedish language, these five languages are somewhat similar and related (Indo-European), but Swedish differs from the other four (which is also reflected on the dendrogram). While Spanish, Catalan, Bulgarian, and Czech are inflectional languages, Swedish is more analytical. However, nouns and adjectives in Swedish are still declined.

4 Conclusion and Future Work

In this paper, we have investigated the typology of errors made by SyntaxNet in parsing the Universal Dependencies test corpora. We have studied three types of errors that occur during parsing: PoS tagging, dependency labeling and attachment errors. Hierarchical clustering of languages based on error frequency data was performed, using the Ward algorithm and the Pearson correlation metric. Clustering was done on multiple datasets: distributions of different types of errors with or without normalizing text register.

Error-based hierarchical clustering, especially when focused on PoS tagging and attachment errors, showed that languages with similar morphological or syntactic structures are usually grouped in the same cluster. That is, the parser makes similar errors when processing languages with similar structural properties. This might indicate that the performance of a universal parser can be improved by taking into account the structural (morphological and syntactic) types of the languages that are to be analyzed. Perhaps learning can be done for multiple similar (related) languages at the same time.

Further research will be aimed at interpreting the neural network models, using the method proposed in (Li et al. 2017). In addition, parameter selection for the components of SyntaxNet (Morpher, Parser Tagger) is also interesting, since the pre-trained models made available by Google have the same parameters for all languages. Tweaking these hyperparameters could improve parsing quality; furthermore, it might

be possible to reveal general patterns in these parameters, if any, for typologically similar languages.

References

- Andor, D., Alberti, C., Weiss, D., Severyn, A., Presta, A., Ganchev, K., Petrov, S., Collins, M.: Globally normalized transition-based neural networks. arXiv preprint [arXiv:1603.06042](https://arxiv.org/abs/1603.06042) (2016)
- Buchholz, S., Marsi, E.: CoNLL-X shared task on multilingual dependency parsing. In: Proceedings of the Tenth Conference on Computational Natural Language Learning, pp. 149–164. Association for Computational Linguistics (2006)
- Charniak, E.: A maximum-entropy-inspired parser. In: Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference, pp. 132–139. Association for Computational Linguistics (2000)
- Collins, M.: Three generative, lexicalised models for statistical parsing. In: Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics, pp. 16–23. Association for Computational Linguistics (1997)
- Collins, M.: Head-driven statistical models for natural language parsing. *Comput. Linguis.* **29**(4), 589–637 (2003)
- Covington, M.A.: A fundamental algorithm for dependency parsing. In: Proceedings of the 39th Annual ACM Southeast Conference, pp. 95–102 (2001)
- Eisner, J.M.: Three new probabilistic models for dependency parsing: an exploration. In: Proceedings of the 16th Conference on Computational Linguistics, vol. 1, pp. 340–345. Association for Computational Linguistics (1996)
- Ferguson, C.A.: Sports announcer talk: syntactic aspects of register variation. *Lang. Soc.* **12**(2), 153–172 (1983)
- Ferrara, K., Brunner, H., Whittemore, G.: Interactive written discourse as an emergent register. *Written Commun.* **8**(1), 8–34 (1991)
- Haegeman, L.: Understood subjects in English diaries. On the relevance of theoretical syntax for the study of register variation. *Multilingua J. Cross Cult. Interlanguage Commun.* **9**(2), 157–199 (1990)
- Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, vol. 1, pp. 423–430. Association for Computational Linguistics (2003)
- Li, J., Monroe, W., Jurafsky, D.: Understanding neural networks through representation erasure. arXiv preprint [arXiv:1612.08220](https://arxiv.org/abs/1612.08220) (2017)
- Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: the penn treebank. *Comput. Linguit.* **19**(2), 313–330 (1993)
- Matthews, P.H.: Syntax. Cambridge Textbooks in Linguistics, pp. 69–75. Cambridge University Press, Cambridge (1981)
- McDonald, R., Lerman, K., Pereira, F.: Multilingual dependency analysis with a two-stage discriminative parser. In: Proceedings of the Tenth Conference on Computational Natural Language Learning, pp. 216–220. Association for Computational Linguistics (2006)
- McDonald, R.T., Nivre, J.: Characterizing the errors of data-driven dependency parsing models. In: EMNLP-CoNLL, pp. 122–131 (2007)
- McDonald, R.T., Nivre, J., Quirkbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, O., Bedini, C., Castelló, N.B., Lee, J.: Universal dependency annotation for multilingual parsing. In: ACL (2), pp. 92–97 (2013)

- Nivre, J.: Dependency grammar and dependency parsing. *MSI Rep.* **51**33(1959), 1–32 (2005)
- Nivre, J., Hall, J., Nilsson, J.: Maltparser: a data-driven parser-generator for dependency parsing. In: Proceedings of LREC, vol. 6, pp. 2216–2219 (2006)
- Nivre, J., de Marneffe, M.C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C.D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., Zeman, D.: Universal dependencies v1: a multilingual treebank collection. In: Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016), pp. 1659–1666, May 2016
- Petrov, S.: Announcing syntaxnet: The world’s most accurate parser goes open source. Google Research Blog, 12 May 2016
- Ward Jr., J.H.: Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.* **58**, 236–244 (1963)

Size vs. Structure in Training Corpora for Word Embedding Models: Araneum Russicum Maximum and Russian National Corpus

Andrey Kutuzov¹ and Maria Kunilovskaya^{2(✉)}

¹ University of Oslo, Oslo, Norway
andreku@ifi.uio.no
² University of Tyumen, Tyumen, Russia
mkunilovskaya@gmail.com

Abstract. In this paper, we present a distributional word embedding model trained on one of the largest available Russian corpora: Araneum Russicum Maximum (over 10 billion words crawled from the web). We compare this model to the model trained on the Russian National Corpus (RNC). The two corpora are much different in their size and compilation procedures. We test these differences by evaluating the trained models against the Russian part of the Multilingual SimLex999 semantic similarity dataset. We detect and describe numerous issues in this dataset and publish a new corrected version. Aside from the already known fact that the RNC is generally a better training corpus than web corpora, we enumerate and explain fine differences in how the models process semantic similarity task, what parts of the evaluation set are difficult for particular models and why. Additionally, the learning curves for both models are described, showing that the RNC is generally more robust as training material for this task.

Keywords: Word embeddings · Web corpora · Semantic similarity

1 Introduction

It is a widespread opinion in machine-learning contexts that more data is more beneficial than careful preprocessing and selection of the existing material. Is this true when it comes to representing the meaning of the words as the function of their contexts? This research aims to find out whether and how the type of corpus used to train a word embedding model affects the quality of the resulting word embeddings for Russian.

Most experimental work on word embeddings is centered around testing available algorithms and their hyperparameters, while the type of corpora used to train the models attracts less attention. At the same time it is reasonable to suggest that the nature of corpus material behind the model should have some bearing on the performance of the latter.

In this paper, we rigorously compare Araneum Russicum (arguably the largest web-harvested corpus for Russian) against a small but carefully balanced and designed Russian National Corpus (further RNC). We train comparable word embedding models on both corpora and intrinsically evaluate them using the existing semantic similarity and relatedness datasets. Additionally, we reveal some problems in the Russian part of the widely used Multilingual SimLex999 semantic similarity evaluation set, and publish the corrected version.

The paper is structured as follows. In Sect. 2 we put our research in the context of the previous work. Section 3 introduces our corpora, the methods used to train the models, and the gold datasets. Section 4 presents the evaluation results. In Sect. 5 we analyze and compare typical errors made by the models, and in Sect. 6 we conclude.

2 Related Work

Distributional semantic models have been studied and used for decades; see [1] for an extensive review. [2] introduced the highly efficient *Continuous skip-gram* (SGNS) and *Continuous Bag-of-Words* (CBOW) algorithms for training predictive distributional models, using dense vectors. The so-called *word embedding* models became a *de facto* standard in the NLP world in the recent years, outperforming state-of-the-art in many tasks [3]. In the present research, we use the SGNS implementation in the *Gensim* library.

The issue of evaluating distributional semantic models has a long history and is a subject of many discussions (including the special *RepEval* workshop). In the presence of a particular downstream task, it is always better to evaluate the model on this task. However, when training a general-purpose model, one has to rely on intrinsic evaluation, using one of the available gold datasets. The main methods of intrinsic evaluation are numerous, with many available datasets for English. However, in this paper, we limit ourselves to measuring correlation of semantic similarity scores with human judgments, which is also the most established one. For more information on the semantic similarity and relatedness task, especially in Russian context, we refer the reader to [4].

The issue of the influence of the training corpora on the performance of word embedding models for Russian was raised in [5]. Among other, they compared the models trained on the RNC and a randomly sampled corpus of Russian web pages, about an order larger than the RNC. They found out that albeit much smaller, the RNC consistently outperformed the web corpus in the performance of the models trained on it. In this research, we move this even further by using the Araneum Maximum web corpus, which is almost two orders larger than the RNC. Additionally, we carefully analyze the errors of the models and their learning curves. We hypothesize that the types of errors can be different, as it is known that the models trained on corpora of different types can reflect different ‘semantic landscapes’ (see, among others, [6]).

3 Resources Used

3.1 Corpora

The corpora we employ are the *Araneum Russicum Maximum* which is a web corpus of Russian presented in [7], and the *RNC*, which is the flagship academic corpus of Russian. The size of the former corpus is ≈ 10000 million words, and the size of the latter is ≈ 200 million words. Both corpora were lemmatized and tagged with Mystem [8], so that each token was transformed into the ‘LEMMA_PoS’ representations. Afterwards, the PoS tags were converted to the Universal PoS tagset [9]. Functional words, non-alphabetic tokens, punctuation and one-word sentences were removed. These corpora were used to train word embeddings models, as described in the next subsection.

3.2 The Training Algorithm

The models were trained using *Continuous Skipgram* algorithm [2], with vector size 600 and a symmetric context window of 2 words to the left and 2 words to the right. In the choice of the window size we considered the known fact that larger windows induce models that are more ‘associative’, while smaller windows induce more ‘functional’ and ‘synonymic’ models, leading to better performance on similarity datasets [10]. As we are going to evaluate our models on similarity sets, we chose the narrow window of 2.

Low-frequency words were discarded, by using the frequency thresholds of 10 and 400 for the RNC and the Araneum respectively, resulting in the RNC model containing vectors for 173 816 words and the Araneum model containing vectors for 196 465 words. We used 15 negative samples in both cases, and no downsampling (as we removed all the stop words from the corpora beforehand). The sentences in the corpora were shuffled prior to the training to avoid the influence of corpus ordering, and we iterated over each corpus 5 times.

3.3 Russian Part of Multilingual SimLex999 as the Evaluation Set

One of the widely used gold standard sets for testing the ability of distributional models to detect semantic similarity is the SimLex999 [11]. It was developed to address numerous issues of the previous datasets. We tested our models’ performance on the Russian part of *Multilingual SimLex999* evaluation set introduced in [12]. It was created by translating the original English set; the similarity of the resulting word pairs was re-evaluated in a crowd-sourcing effort. We further refer to this set as *RuSimLex999*. It contains word pairs and the corresponding similarity values (for example, *мудрость-ум* ‘*wisdom-intellect*’ 8.23). We tagged the words in the set with the Universal PoS tags, using Mystem and respecting the dataset section the word belonged to (*RuSimLex999* consists of separate sections on nouns, adjectives and verbs).

Despite its popularity, the manual inspection of original *RuSimLex999* revealed numerous technical flaws. Therefore, we produced an improved version of this dataset by resolving the issues noticed. In this paper we report the

performance of our models on this corrected set *RuSimLex965* along with the original *RuSimLex999*. The issues addressed are as follows:

1. 18 duplicate word pairs (one word pair repeated across the dataset):
 - six pairs with equal scores (e.g. брать-получать 4.08 ‘take-receive’) were deduplicated and only one copy was retained in the revised set,
 - nine duplicate pairs with different scores (e.g. принимать-отвергать 0/0.69 ‘accept-reject’) - we assigned the average score for all duplicate pairs to the unique pair retained in the set,
 - three pairs consisting of the same words in the reversed order with different scores (e.g. работодатель-работник 2.08 ‘employer-employee’; работник-работодатель 1.77 ‘worker-employer’) - we retained one pair with the average score for all duplicate pairs,
 - one pair containing two identical words (палец-палец 9.92 ‘finger-finger’; most likely from the English ‘toe-finger’) was deleted from the set;
2. two pairs which are hardly adequate due to frequency concerns: for example, рапшкуль-уголь 1.38 (*charcoal-coal*). The first word never occurs in the RNC and is found only five times in the Araneum, which means its IPM is as low as 0.0005. It is unlikely that untrained native speakers are able to quantify the difference in this case consistently and yet it is one of the criteria to be met by a gold standard [11]. It is not surprising that this word along with бедствование (*calamitousness*) is unknown to both our models. All the other words from the dataset are covered by the models;
3. there are several typos in the set. We had to correct spelling in three pairs (путешествие-з**ав**ование, 0.69; приворяться-казаться, 4.92; отсутствие-пристуствие, 0.08)
4. in the pair мука-горе 0.277 (*torment-grief*) there is an unnecessary ambiguity: мука can mean both ‘*torment*’ and ‘*flour*’ in Russian, depending on the stress. The original English SimLex999 pair is ‘*agony-grief*’, but the correct reading is arguably not the first coming to the mind of the Russian speaker. This pair was deleted from the revised set;
5. many translations could have been more adapted: the Russian test set contains many loan words (джет, цент, доллар, бренди ‘*cent, brandy*’) at the same time lacking the respective Russian words рубль, водка (‘*ruble, vodka*’).

It is noteworthy that the original English SimLex999 is free of the above shortcomings. In the German translation, there is one duplicate pair with different scores (*schlecht, schrecklich*); the Italian counterpart had four duplicate pairs, including one with the same scores (*felice, arrabbiato*).

With PoS consistency in mind, we had to further delete 12 word pairs that were impossible to annotate in accordance with the evaluation set logic (these pairs when tagged include different parts of speech, which might affect the model performance). In three more pairs we had to adjust lemma tags to ensure homogeneity of the set. For example, the default out-of-context tagging, which returned the pair ранить_VERB-бесстрастный_ADJ (literary ‘*to injure-unemotional*’; most likely from ‘*fragile-frigid*’), was changed to

the intended `ранимый_ADJ-бесстрастный_ADJ` ('*vulnerable-unemotional*'), because `ранимый_ADJ` ('*vulnerable*') was frequent in the tagged corpora. After tagging, we also had to fix some lemmatization errors such as `виски` ('*whisky*'), which was lemmatized as `висок` ('*temple*') or `легкое` ('*lungs*') paired with '*liver*', which was lemmatized to the Russian lemma `легкий` ('*easy or light*').

Thus, the cleaned and improved *RuSimLex965* semantic similarity evaluation test set includes 965 unique word pairs, consistently PoS-tagged and congruent with the Mystem output.

3.4 RuSSE Evaluation Sets

For comparison, we also report results for 3 evaluation sets described in [4]:

1. RuSSE HJ; translated to Russian from the widely used datasets for English;
2. RuSSE WS353 Similarity; translated from the *WS353* dataset [13];
3. RuSSE WS353 Relatedness; the same source.

These alternative sets were produced in the context of RuSSE, the first semantic similarity shared task for Russian. In all three datasets, the scores were obtained by a crowd-sourcing initiative employing native Russian speakers.

Unlike *RuSimLex999*, these sets include only nouns with the exception of one pair `подписать-перерыв` ('*to sign*'-'*a break*') 0.0. Other internal discrepancies of the sets include a pair consisting of two identical words '`тигр-тигр`' ('*tiger*') 0.875, two duplicate pairs '`приспособление-инструмент`' ('*appliance-tool*') 0.708/0.615 and a pair with a non-Cyrillic non-word '`фонд-cd`' ('*fund-cd*') 0. Agian, we PoS-tagged these sets with Mystem. After filtering and preprocessing the full HJ set contains 394 word pairs, the WS353 similarity component contains 248 pairs and WS353 relatedness component contains 199 pairs.

Note that the HJ dataset is produced from several different English sets (most pairs come from *WS353*) and does not distinguish between semantic similarity and relatedness. The separate *WS353-sim* and *WS353-rel* test sets are more consistent; however, they are much smaller than *RuSimLex999*, contain only nouns and still suffer from the shortcomings identified in [11]. It is also important that *RuSimLex999* features significantly higher inter-rater agreement than the RuSSE HJ data set (Krippendorf's alpha 0.57 and 0.49 respectively). Because of these factors, we chose *RuSimLex999* as our primary evaluation measure, despite its flaws described above.

4 Results

For both models we measured Spearman correlation between the similarities produced by the models and the scores provided in the datasets. In the two cases of out-of-vocabulary word pairs for the original dataset, 0.0 was used as a placeholder for the model similarity. Table 1 presents the results. All the scores are statistically significant, with p value well below 0.01 level.

Table 1. Spearman’s ρ for models scores correlation against gold datasets.

Corpus	RuSSE sets				
	RuSimLex999	RuSimLex965	HJ	WS353-Sim	WS353-rel
RNC	43.22	42.55	70.69	74.11	59.07
Araneum	42.52	41.46	73.38	77.51	63.15

The RNC-based models consistently outperform the Araneum-based ones on the SimLex datasets, but lose when evaluated against the RuSSE datasets. We suppose that the reason for this is that the RuSSE sets are not as rigorous in distinguishing semantic relatedness (can the word a be substituted with the word b ?) versus semantic similarity (is the word a associated with the word b ?). *WS353-Sim* contains only similar pairs and *WS353-Rel* contains only related pairs (HJ incorporates both, together with several other smaller test sets).

Consider the pair ‘день-рассвет’ (*‘day-sunrise’*). In *RuSimLex999*, its score is only 1.54 (rank 655 out of 1 000), thus this pair elements are quite far away from each other. At the same time, in *WS353-rel*, the same pair features score 4.44 and is ranked 67 out of 249, meaning that these two words are really close. In *WS353-sim*, there are no related pairs at all.

Thus, the large Araneum corpus provides better training data to properly rank *either* related or similar pairs. However, when faced with the task to rank similar pairs higher than simply associated ones, it shows up as inferior to the RNC. It means that more data helps only when the downstream task allows to not care for one of the closeness types. If, on the other hand, one needs to clearly rank the ‘coffee-americano’ pair higher than the ‘coffee-cup’, smaller but balanced corpora pay off. Also, the corrected *RuSimLex965* seems to be a bit more difficult for the models than the original one.

Table 2. Correlations for different PoS subsets of *RuSimLex965*.

PoS	Araneum	RNC	Number of pairs
Nouns	41.67	43.49	653
Adjectives	47.92	42.31	97
Verbs	44.20	44.65	215

If we measure the correlation on separate subsets consisting of pairs belonging to one and the same PoS, some interesting differences can be observed. Table 2 presents these scores. For nouns and verbs, the RNC model is better, and for both models verbs are easier. Unexpected results show up for adjectives, which seem to be most difficult for the RNC models, but the easiest for the Araneum one: actually, this is the only PoS-subset on which the Araneum model *outperforms* the RNC-based one, for reasons unclear. Note also that the fact that the RNC

model is better with nouns on *RuSimLex965* does not prevent it from losing on the RuSSE sets which contain only nouns. This again proves that these sets feature different kind of similarity scores, arguably mixed with relatedness.

4.1 Learning Curves

We additionally studied how fast the models were in achieving a good performance, as we added more data. To this end, we trained models on the incrementally increased ‘slices’ of our corpora (e.g., the first 10 million words, then the first 20 million words, etc.). For all these models the frequency threshold hyperparameter was set to 10. The results are shown in the Fig. 1.

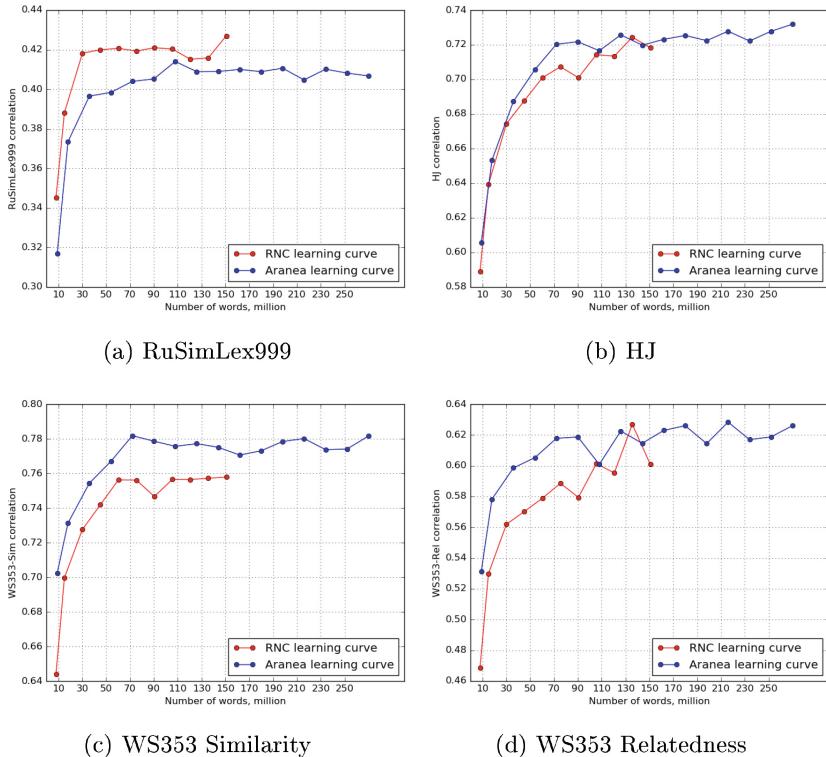


Fig. 1. Learning curves for Araneum-based and RNC-based models.

For *RuSimLex999*, the RNC provides much better training data from the very start. Even the models trained on the first 10 million words of both corpora already differ by 5 points. Further on, as we train the models on more and more text data, this difference is preserved: the RNC model consistently outperforms the Araneum model, and never vice versa. If we evaluate Araneum trained on approximately the same amount of data as the whole RNC, its *RuSimLex999* performance will achieve only 0.41 (0.43 for the RNC).

The models show similar patterns of development as they are fed with more data. Particularly, moving from 10 to 15..18 million training words makes a huge difference, as well as moving further on to 30 million. After that, the performance on *RuSimLex999* stabilizes and improvements become much smaller. Another interesting discovery is that sometimes after adding more data, the performance drops for a while. It happens almost simultaneously with the RNC and the Araneum models near 110 million words mark. However, after another 20 million words the models overcome this drop and return to gradual improving.

On the RuSSE test sets, the Araneum models outperform the RNC ones from the very beginning as well. However, except for the *WS353-Sim* set, the results are unstable, with the RNC achieving comparable performance at some times, or even outperforming the Araneum model. Interestingly, on the *HJ* and *WS353-Rel* sets, the RNC model achieves comparable performance at approximately the same moment (about 110 million words) when the Araneum model comes closest to the RNC one on the *RuSimLex999*. This further supports our hypothesis that these sets are complementary. At the same time, *WS353-Sim* should in theory be similar to *RuSimLex999*, but in practice it is the most consistent in showing Araneum outperforming RNC. We have not come to any final conclusion about the reasons for this behavior and leave this for future work.

5 Error Analysis: What the Models Do Wrong

5.1 Model-Specific Errors

To compare performance of the models based on the two corpora, we analyzed each model's distinctive errors against the gold standard and relatively more adequate judgment of the competing model. To arrive at the list of these errors for each model we took the following steps:

1. ranked pairs by their similarity scores produced by the two models and by human raters; we had to assign pairs with the same score the same ranks;
2. determined the delta between the ranks of word pairs in the descending list of similarity scores for each model and the gold standard (*RuSimLex965*) (columns 9 and 11 in Table 3);
3. sorted the deltas in the descending order and determined quartiles in the sort: Q1 represents pairs whose model score is very different from the gold one; the lower the quartile, the more accurate the model's assessment is;
4. distinctive errors of each model against each other are pairs with contrasting ranks in the lists above; at their strongest, they come from opposite quartiles. We found the differences between the quartile numbers for each pair and sorted the list in descending order (column 13).

Table 3 shows the top three and bottom three rows in the results. To make our reasoning explicit let us consider the first example from Table 3. The rank difference between Araneum and *RuSimLex965* for the hyponym/hyperonym pair ‘кот-питомец’ (‘cat-pet’) is twice smaller (234) than for RNC (435). In this

case Araneum overestimates similarity placing the pair higher in the ranking (122 is higher than 356), while the RNC underestimates similarity the pair is way below where it should be from the gold standard point of view (791 instead of around 356). The pair belongs to Quartile 4 (Araneum) and Quartile 1 (RNC) of the respective descending lists of absolute rank differences. The value in column 13 (3) indicates that the RNC similarity estimation for this pair is very distant from the gold standard and Araneum.

Table 3. Models errors against *RuSimLex965* and the competing model.

No	Word pair	Similarity scores			Ranks			Rank differences and Quartiles				Diff
		Araneum	RNC	SimLex	Araneum	RNC	SimLex	Δ Ar.-SimLex Ranks	Quart	Δ RNC-SimLex Ranks	Quart	
1	2	3	4	5	6	7	8	9	10	11	12	13
1.	кот-питомец ‘cat-pet’	0.56	0.22	0.45	122	791	356	234	4	435	1	3
2.	витамин-железо ‘vitamin-iron’	0.38	0.23	0.27	420	773	511	91	3	262	1	2
3.	аргументировать- подтверждать ‘argue - justify’	0.35	0.24	0.55	454	734	270	184	3	464	1	2
...												
963.	позволять- разрешать ‘allow-permit’	0.21	0.41	0.94	767	289	16	751	1	273	3	-2
964.	скрипка- инструмент ‘violin-instrument’	0.32	0.44	0.45	535	225	356	179	2	131	4	-2
965.	трубка-сигара ‘pipe-cigar’	0.33	0.44	0.53	509	224	284	225	2	60	4	-2

The full content of Table 3 gives a general overview of how the machine judgments compare to each other with respect to distance from the gold standard. The top 12 and bottom 5 word pairs have the interquartile difference of 2 or more. They represent the largest relative errors (given the judgment of the competing model) for the RNC and Araneum respectively. Further top 162 and bottom 165 pairs have the difference of 1. Note that most of the 965 pairs we tested received the similar scores for Araneum and the RNC, signaling common learning potential of the two corpora. In fact, in our experiment 638 word pairs (66% of the set) ended up in the same quartile for both models.

Below we focus on the 101 word pairs (10%) for which the performance of the models differs the most. The top 44 pairs (where RNC errs more than Araneum) are dominated by: synonyms (18 pairs), pairs with high levels of association based on contiguity of referents in reality or domain relatedness of the words (14 pairs), and hyponymy/cohyponymy/hyperonymy pairs, particularly verbal (8 pairs).

Interestingly, only 11 pairs out of 44 have a negative rank difference with the gold standard, that is only 11 pairs are placed higher in the ranking than they are in *RuSimLex965*. In two-thirds of the cases the RNC model underestimates similarity in contrast with human scores and almost correct Araneum judgment.

The same semantic groups are prevalent at the bottom of Table 3, among 57 word pairs for which Araneum gives erroneously higher or lower scores, while RNC gets them almost right (based on rank difference again, not raw scores).

The ratio of the groups is more tilted towards unrecognized synonyms, however. Another contrast is in processing strongly associated pairs and different types of hyponymy. While RNC tends to erroneously downplay similarity of concepts related by association or as members of the same classification (general-specific), Araneum places them higher in rank than both the RNC and *RuSimLex965*. The comparison is presented in Table 4.

Table 4. Semantic relations of word pairs with the greatest discrepancy in ranks between each model and *RuSimLex965*/competing model

Relation	RNC		Araneum		Examples (from Araneum model rankings)
	under	over	under	over	
synonymy	16	2	23	2	твёрдый-прочный ‘hard-tough’
association	9	5	5	8	кровь-плоть ‘blood-flesh’
hyponymy	7	1	2	7	гитара-барабан ‘guitar-drum’
meronymy	1	2	2	1	кость-локоть ‘bone-elbow’
antonymy	0	1	1	4	работник-работодатель ‘worker-employee’
missing value	0	0	1	0	ребячливый-безрассудный ‘childish-foolish’
non-similar	0	0	0	1	дорожка-шар ‘path-ball’
Total	33	11	34	23	

The analysis shows that both models have difficulties recognizing synonyms. Synonyms constitute by far the largest group of underestimation errors. One of the possible reasons can be simply that there are many synonymous pairs in the test sets, and thus a large portion of them is low-frequency words which means their embeddings are not perfect.

According to [11], ‘similarity is a cognitively complex operation that can require rich, structured conceptual knowledge to compute accurately’. We did not find factors that affected the models’ performance in processing synonyms (such as type of synonymy or their part of speech). Interestingly, two of four pairs whose similarity was overestimated had unreasonably low similarity scores in *RuSimLex965*: верование-мнение (‘belief-opinion’) 0.131 and друг-парень (‘friend-buddy’) 0.1. This again poses a question of the evaluation set quality.

There are slight differences in how the two models represent association. The model based on Araneum is likely to overestimate similarity in this case, while the RNC model errs on the underestimate side in the same cases. This partly explains high performance of the Araneum models on the *WS353-Rel* test set.

Another group of semantic relations that constitute subtypes of similarity (not association) includes hyponymy/hyperonymy and cohyponymy. This is the only single semantic relation with a stark contrast between the two models. For hyponym/hyperonym pairs, the Araneum model learns vectors that return high similarity scores, while this type of similarity goes widely unrecognized by the

RNC model (whether this is good, depends on one’s particular task, but it seems to be favored by the RuSSE test sets).

A further finding is that the model trained on Araneum is misled by antonymy. Antonyms are known to share syntactic distributions and therefore, get high semantic similarity scores in distributional models, while humans have no difficulty assigning low similarities to different types of opposites (binaries, gradual/directional, non-binary, relational opposites). Our experiments show that both models overestimate similarity of antonyms, but Araneum performs comparatively worse. All five pairs of antonyms that made it to the list of the Araneum model errors are the so called relative antonyms. Their meanings are opposed given a particular relation between entities or a special situation they are usually part of: вода-дед, тетя-племянник, работодатель-работник, терять-подучать (‘water-ice’, ‘aunt-nephew’, ‘employer-employee’, ‘lose-gain’).

6 Conclusion

We compared the performance of Continuous Skipgram word embedding models trained on the very large web corpus of Russian (Araneum Maximum) and the much smaller Russian National Corpus. As our primary evaluation set, we chose the Russian part of Multilingual SimLex999. We revealed numerous flaws in its design and eventually came up with its refined version, *RuSimLex965*. We publish its raw and PoS-tagged variants¹, together with the trained models² and the tagged Araneum Maximum corpus³. Note that there are still some conceptual issues in the SimLex999 test set (cf. [14]). Also, all these datasets were originally compiled in English and then re-scored by native Russian speakers, which may degrade their reliability. To address these problems one has to compile a new dataset from scratch, which is outside the scope of the present research.

With both variants of the evaluation set, our experiments supported the previous work in that a balanced national corpora, albeit smaller, consistently outperform large web-based corpora in semantic similarity evaluation setting. At the same time, the Araneum-based model was superior on the sets containing semantic relatedness scores; thus, this corpus is more suitable for calculating associative, topical and hyponymic relations between words.

Further, we analyzed the speed of performance improving with increasing the size of the training data for both corpora. We show that almost all improvement stops after the first 100 million words for semantic similarity test sets, and that the RNC ‘saturates’ somewhat faster than the Araneum. For the semantic relatedness test sets, it seems that the model performance does not saturate and continues to improve after this point as well. Finally, we performed an extensive error analysis for both models, revealing typical classes of errors, and how the RNC and Araneum models differ in this respect.

¹ <http://rusvectores.org/static/testsets/>.

² <http://rusvectores.org/models/>.

³ http://rusvectores.org/static/rus_araneum_maxicum.txt.gz.

As a future work, we plan to study in more detail how different are RuSSE test sets from the *RuSimLex999*. We also would like to find out whether our findings hold for English and other languages, as well as for other types of intrinsic evaluation (analogical inference, etc.).

References

1. Turney, P., Pantel, P., et al.: From frequency to meaning: vector space models of semantics. *J. Artif. Intell. Res.* **37**(1), 141–188 (2010)
2. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **26**, 3111–3119 (2013)
3. Baroni, M., Dinu, G., Kruszewski, G.: Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol. 1, Baltimore, USA, pp. 238–247 (2014)
4. Panchenko, A., Ustalov, D., Arefyev, N., Paperno, D., Konstantinova, N., Loukachevitch, N., Biemann, C.: Human and machine judgements about Russian semantic relatedness. In: Proceedings of the 5th Conference on Analysis of Images, Social Networks and Texts (AIST 2016), Communications in Computer and Information Science (CCIS), pp. 174–183 (2016)
5. Kutuzov, A., Andreev, I.: Texts in, meaning out: neural language models in semantic similarity task for Russian. In: Proceedings of the Dialog Conference, vol. 2, Moscow, Russia, pp. 133–145 (2015)
6. Kutuzov, A., Kuzmenko, E.: Comparing neural lexical models of a classic national corpus and a web corpus: the case for russian. In: Gelbukh, A. (ed.) CICLing 2015. LNCS, vol. 9041, pp. 47–58. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18111-0_4
7. Benko, V., Zakharov, V.: Very large Russian corpora: new opportunities and new challenges. *Kompyuternaja Lingvistika I Intellektuanyje Technologii: Po Materialam Medunarodnoj konferencii "Dialog"*, vol. 15(22), pp. 79–93 (2016)
8. Segalovich, I.: A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. In: MLMTA, pp. 273–280 (2003)
9. Petrov, S., Das, D., McDonald, R.: A universal part-of-speech tagset. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), European Language Resources Association (ELRA) (2012)
10. Goldberg, Y.: A primer on neural network models for natural language processing. *J. Artif. Intell. Res.* **57**, 345–420 (2016)
11. Hill, F., Reichart, R., Korhonen, A.: Simlex-999: evaluating semantic models with (genuine) similarity estimation. *Comput. Linguist.* **41**(4), 665–695 (2015)
12. Leviant, I., Reichart, R.: Separated by an un-common language: towards judgment language informed vector space modeling. arXiv preprint [arXiv:1508.00106](https://arxiv.org/abs/1508.00106) (2015)
13. Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E.: Placing search in context: the concept revisited. In: Proceedings of the 10th International Conference on World Wide Web, pp. 406–414. ACM (2001)
14. Avraham, O., Goldberg, Y.: Improving reliability of word similarity evaluation by redesigning annotation task and performance measure. In: Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP. Association for Computational Linguistics, pp. 106–110 (2016)

Combining Thesaurus Knowledge and Probabilistic Topic Models

Natalia Loukachevitch^{1(✉)}, Michael Nokel², and Kirill Ivanov¹

¹ Lomonosov Moscow State University, Moscow, Russia

louk_nat@mail.ru, ivanov.kir.m@yandex.ru

² Yandex, Moscow, Russia

mnokel@gmail.com

Abstract. In this paper we present the approach of introducing thesaurus knowledge into probabilistic topic models. The main idea of the approach is based on the assumption that the frequencies of semantically related words and phrases, which are met in the same texts, should be enhanced: this action leads to their larger contribution into topics found in these texts. We have conducted experiments with several thesauri and found that for improving topic models, it is useful to utilize domain-specific knowledge. If a general thesaurus, such as WordNet, is used, the thesaurus-based improvement of topic models can be achieved with excluding hyponymy relations in combined topic models.

Keywords: Thesaurus · Multiword expression
Probabilistic topic models

1 Introduction

Currently, probabilistic topic models are important tools for improving automatic text processing including information retrieval, text categorization, summarization, etc. Besides, they can be useful in supporting expert analysis of document collections, news flows, or large volumes of messages in social networks [1–3]. To facilitate this analysis, such approaches as automatic topic labeling and various visualization techniques have been proposed [2,5].

Boyd-Graber et al. [4] indicate that to be understandable by humans, topics should be specific, coherent, and informative. Relationships between the topic components can be inferred. In [2] four topic visualization approaches are compared. The authors of the experiment concluded that manual topic labels include a considerable number of phrases; users prefer shorter labels with more general words and tend to incorporate phrases and more generic terminology when using more complex network graph. Blei and Lafferty [5] visualize topics with ngrams consisting of words mentioned in these topics. These works show that phrases and knowledge about hyponyms/hypernyms are important for topic representation.

In this paper we describe an approach to integrate large manual lexical resources such as WordNet or EuroVoc into probabilistic topic models, as well

as automatically extracted n-grams to improve coherence and informativeness of generated topics. The structure of the paper is as follows. In Sect. 2 we consider related works. Section 3 describes the proposed approach. Section 4 enumerates automatic quality measures used in experiments. Section 5 presents the results obtained on several text collections according to automatic measures. Section 6 describes the results of manual evaluation of combined topic models for Islam Internet-site thematic analysis.

2 Related Work

Topic modeling approaches are unsupervised statistical algorithms that usually considers each document as a “bag of words”. There were several attempts to enrich word-based topic models (=unigram topic models) with additional prior knowledge or multiword expressions.

Andrzejewski et al. [6] incorporated knowledge by Must-Link and Cannot-Link primitives represented by a Dirichlet Forest prior. These primitives were then used in [7], where similar words are encouraged to have similar topic distributions. However, all such methods incorporate knowledge in a hard and topic-independent way, which is a simplification since two words that are similar in one topic are not necessarily of equal importance for another topic.

Xie et al. [8] proposed a Markov Random Field regularized LDA model (MRF-LDA), which utilizes the external knowledge to improve the coherence of topic modeling. Within a document, if two words are labeled as similar according to the external knowledge, their latent topic nodes are connected by an undirected edge and a binary potential function is defined to encourage them to share the same topic label. Distributional similarity of words is calculated beforehand on a large text corpus.

In [9], the authors gather so-called lexical relation sets (LR-sets) for word senses described in WordNet. The LR-sets include synonyms, antonyms and adjective-attribute related words. To adapt LR-sets to a specific domain corpus and to remove inappropriate lexical relations, the correlation matrix for word pairs in each LR-set is calculated. This matrix at the first step is used for filtering inappropriate senses, then it is used to modify the initial LDA topic model according to the generalized Polya urn model described in [10]. The generalized Polya urn model boosts probabilities of related words in word-topic distributions.

Gao and Wen [11] presented Semantic Similarity-Enhanced Topic Model that accounts for corpus-specific word co-occurrence and word semantic similarity calculated on WordNet paths between corresponding synsets using the generalized Polya urn model. They apply their topic model for categorizing short texts.

All above-mentioned approaches on adding knowledge to topic models are limited to single words. Approaches using ngrams in topic models can be subdivided into two groups. The first group of methods tries to create a unified probabilistic model accounting unigrams and phrases. Bigram-based approaches include the Bigram Topic Model [12] and LDA Collocation Model [13]. In [14] the Topical N-Gram Model was proposed to allow the generation of ngrams

based on the context. However, all these models are enough complex and hard to compute on real datasets.

The second group of methods is based on preliminary extraction of ngrams and their further use in topics generation. Initial studies of this approach used only bigrams [15, 16]. Nokel and Loukachevitch [17] proposed the LDA-SIM algorithm, which integrates top-ranked ngrams and terms of information-retrieval thesauri into topic models (thesaurus relations were not utilized). They create similarity sets of expressions having the same word components and sum up frequencies of similarity set members if they co-occur in the same text.

In this paper we describe the approach to integrate whole manual thesauri into topic models together with multiword expressions.

3 Approach to Integration Whole Thesauri into Topic Models

In our approach we develop the idea of [17] that proposed to construct similarity sets between ngram phrases between each other and single words. Phrases and words are included in the same similarity set if they have the same component word, for example, *weapon – nuclear weapon – weapon of mass destruction; discrimination – racial discrimination*. It was supposed that if expressions from the same similarity set co-occur in the same document then their contribution into the document’s topics is really more than it is presented with their frequencies, therefore their frequencies should be increased. In such an approach, the algorithm can “see” similarities between different multiword expressions with the same component word.

In our approach, at first, we include related single words and phrases from a thesaurus such as WordNet or EuroVoc in these similarity sets. Then, we add preliminarily extracted ngrams into these sets and, this way, we use two different sources of external knowledge. We use the same LDA-SIM algorithm as described in [17] but study what types of semantic relations can be introduced into such similarity sets and be useful for improving topic models. The pseudocode of LDA-SIM algorithm is presented in Algorithm 1, where $S = \{S_w\}$ is a similarity set, expressions in similarity sets can comprise single words, thesaurus phrases or generated noun compounds.

We can compare this approach with the approaches applying the generalized Polya urn model [9–11]. To add prior knowledge, those approaches change topic distributions for related words globally in the collection. We modify topic probabilities for related words and phrases locally, in specific texts, only when related words (phrases) co-occur in these texts.

4 Automatic Measures to Estimate the Quality of Topic Models

To estimate the quality of topic models, we use two main automatic measures: topic coherence and kernel uniqueness. For human content analysis, measures of

Algorithm 1. LDA-SIM algorithm

Input: collection D , vocabulary W , number of topics $|T|$, initial $\{p(w|t)\}$ and $\{p(t|d)\}$, sets of similar expressions S , hyperparameters $\{\alpha_t\}$ and $\{\beta_w\}$, n_{dw} is the frequency of w in the document d

Output: distributions $\{p(w|t)\}$ and $\{p(t|d)\}$

```

1 while not meet the stop criterion do
2   for  $d \in D, w \in W, t \in T$  do
3      $p(t|d, w) = \frac{p(w|t)p(t|d)}{\sum_{u \in T} p(w|u)p(u|d)}$ 
4   for  $d \in D, w \in W, t \in T$  do
5      $n'_{dw} = n_{dw} + \sum_{s \in S_w} n_{ds}$ 
6      $p(w|t) = \frac{\sum_{d \in D} n'_{dw} p(t|d, w) + \beta_w}{\sum_{d \in D} \sum_{w \in d} n'_{dw} p(t|d, w) + \sum_{w \in W} \beta_w}$ 
7      $p(t|d) = \frac{\sum_{w \in d} n'_{dw} p(t|d, w) + \alpha_t}{\sum_{w \in W} \sum_{t \in T} n'_{dw} p(t|d, w) + \sum_{t \in T} \alpha_t}$ 

```

topic coherence and kernel uniqueness are both important and complement each other. Topics can be coherent but have a lot of repetitions. On the other hand, generated topics can be very diverse, but incoherent within each topic.

Topic coherence is an automatic metric of interpretability. It was shown that the coherence measure has a high correlation with the expert estimates of topic interpretability [10, 19]. Mimno et al. [10] described an experiment comparing expert evaluation of LDA-generated topics and automatic topic coherence measures. It was found that most “bad” topics consisted of words without clear relations between each other.

Newman et al. [7] asked users to score topics on a 3-point scale, where 3 = “useful” (coherent) and 1 = “useless” (less coherent). They instructed the users that one indicator of usefulness is the ease by which one could think of a short label to describe a topic. Then several automatic measures, including WordNet-based measures and corpus co-occurrence measures, were compared. It was found that the best automatic measure having the largest correlation with human evaluation is word co-occurrence calculated as point-wise mutual information (PMI) on Wikipedia articles. Later Lau et al. [19] showed that normalized pointwise mutual information (NPMI) [20] calculated on Wikipedia articles correlates even more strongly with human scores.

We calculate automatic topic coherence using two measure variants. The coherence of a topic is the median PMI (NPMI) of word pairs representing the topic, usually it is calculated for n most probable elements (in our study ten elements) in the topic. The coherence of the model is the median of the topic coherence. To make this measure more objective, it should be calculated on an external corpus [19]. In our case, we use Wikipedia dumps.

$$PMI(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \quad NPMI(w_i, w_j) = \frac{PMI(w_i, w_j)}{-\log(p(w_i, w_j))} \quad (1)$$

Human-constructed topics usually have unique main words. The measure of kernel uniqueness shows to what extent topics are different from each other and is calculated as the number of unique elements among most probable elements of topics (kernels) in relation to the whole number of elements in kernels.

$$U(\Phi) = \frac{|\cup_t kernel(T_i)|}{\sum_{t \in T} |kernel(T_i)|} \quad (2)$$

If uniqueness of the topic kernels is closer to zero then many topics are similar to each other, contain the same words in their kernels. In this paper the kernel of a topic means the ten most probable words in the topic. We also calculated perplexity as the measure of language models. We use it for additional checking the model quality.

5 Use of Automatic Measures to Assess Combined Models

For evaluating topics with automatic quality measures, we used several English text collections and one Russian collection (Table 1). We experiment with three thesauri: WordNet¹ (155 thousand entries), information-retrieval thesaurus of the European Union EuroVoc (15161 terms)², and Russian thesaurus RuThes (115 thousand entries)³ [18].

Table 1. Text collections for experiments

Text collection	Number of texts	Number of words
English part of Europarl corpus	9672	≈56 mln
English part of JRC-Acquist corpus	23545	≈53 mln
ACL Anthology Reference corpus	10921	≈48 mln
NIPS Conference Papers (2000–2012)	17400	≈5 mln
Russian banking texts	10422	≈32 mln

At the preprocessing step, documents were processed by morphological analyzers. Also, we extracted noun groups as described in [17]. As baselines, we use the unigram LDA topic model and LDA topic model with added 1000 ngrams with maximal NC-value [21] extracted from the collection under analysis.

As it was found before [15, 17], the addition of ngrams without accounting relations between their components considerably worsens the perplexity because of the vocabulary growth (for perplexity the less is the better) and practically does not change other automatic quality measures (Table 2).

¹ <https://wordnet.princeton.edu/>.

² <http://eurovoc.europa.eu/drupal/>.

³ http://www.labinform.ru/pub/ruthes/index_eng.htm.

Table 2. Integration of WordNet into topic models

Collection	Method	TC-PMI	TC-NPMI	Kernel Uniq	Perplex.
Europarl	LDA unigram	1.20	0.24	0.33	1466
	LDA+1000ngram	1.19	0.23	0.35	2497
	LDA-Sim+WNsyn	1.05	0.26	0.16	1715
	LDA-Sim+WNsynrel	1.20	0.25	0.18	4984
	LDA-Sim+WNsr/hyp	1.47	0.24	0.33	1502
	LDA-Sim+WNsr/hyp+Ngrams	2.08	0.23	0.42	1929
	LDA-Sim+WNsr/hyp+Ngrams/1	2.46	0.25	0.43	1880
JRC	LDA unigram	1.42	0.24	0.53	807
	LDA+1000ngrams	1.46	0.22	0.56	1140
	LDA-Sim+WNsyn	1.32	0.25	0.44	854
	LDA-Sim+WNsynrel	1.26	0.27	0.28	1367
	LDA-Sim+WNsynrel/hyp	1.57	0.24	0.54	823
	LDA-Sim+WNsr/hyp+Ngrams	1.54	0.19	0.64	1093
	LDA-Sim+WNsr/hyp+Ngrams/1	1.58	0.18	0.68	1064
ACL	LDA unigram	1.63	0.24	0.51	1779
	LDA+1000ngrams	1.55	0.23	0.51	2277
	LDA-Sim+WNsyn	1.42	0.26	0.47	1853
	LDA-Sim+WNsynrel	1.26	0.27	0.35	2554
	LDA-Sim+WNsynrel/hyp	1.56	0.24	0.51	1785
	LDA-Sim+WNsr/hyp+Ngrams	2.72	0.28	0.69	2164
	LDA-Sim+WNsr/hyp+Ngrams/1	3.04	0.28	0.76	2160
NIPS	LDA unigram	1.60	0.24	0.41	1284
	LDA+1000ngrams	1.54	0.24	0.41	1969
	LDA-Sim+WNsyn	1.34	0.26	0.39	1346
	LDA-Sim+WNsynrel	1.20	0.27	0.29	2594
	LDA-Sim+WNsynrel/hyp	1.78	0.25	0.43	1331
	LDA-Sim+WNsr/hyp+Ngrams	3.18	0.31	0.62	1740
	LDA-Sim+WNsr/hyp+Ngrams/1	3.27	0.30	0.67	1741

We add the Wordnet data in the following steps. At the first step, we include WordNet synonyms (including multiword expressions) into the proposed similarity sets (LDA-Sim+WNsyn). At this step, frequencies of synonyms found in the same document are summed up in process LDA topic learning as described in Algorithm 1. We can see that the kernel uniqueness becomes very low, topics are very close to each other in content (Table 2: LDA-Sim+WNsyn). At the second step, we add word direct relatives (hyponyms, hypernyms, etc.) to similarity sets. Now the frequencies of semantically related words are added up enhancing the contribution into all topics of the current document.

The Table 2 shows that these two steps lead to great degradation of the topic model in most measures in comparison to the initial unigram model: uniqueness of kernels abruptly decreases, perplexity at the second step grows by several

Table 3. Integration of EuroVoc into topic models

Collection	Method	TC-PMI	TC-NPMI	Kernel Uniq	Perplex.
Europarl	LDA unigram	1.20	0.24	0.33	1466
	LDA+1000ngram	1.19	0.23	0.35	2497
	LDA-Sim+EVsyn	1.57	0.24	0.43	1655
	LDA-Sim+EVsynrel	1.39	0.24	0.35	1473
	LDA-Sim+EVsr/hyp+Ngrams	2.51	0.26	0.50	1957
	LDA-Sim+EVsr/hyp+Ngrams/l	2.5	0.25	0.45	1882
JRC	LDA unigram	1.42	0.24	0.53	807
	LDA+1000ngrams	1.46	0.22	0.56	1140
	LDA-Sim+EVsyn	1.65	0.25	0.57	857
	LDA-Sim+EVsynrel	1.71	0.24	0.57	844
	LDA-Sim+EVsr/hyp+Ngrams	1.91	0.21	0.68	1094
	LDA-Sim+EVsr/hyp+Ngrams/l	1.5	0.18	0.67	1061

times (Table 2: LDA-Sim+WNsynrel). It is evident that at this step the model has a poor quality. When we look at the topics, the cause of the problem seems to be clear. We can see the overgeneralization of the obtained topics. The topics are built around very general words such as “person”, “organization”, “year”, etc. These words were initially frequent in the collection and then received additional frequencies from their frequent synonyms and related words.

Then we suppose that these general words were used in texts to discuss specific events and objects, therefore, we change the constructions of the similarity sets in the following way: we do not add word hyponyms to its similarity set. Thus, hyponyms, which are usually more specific and concrete, should obtain additional frequencies from upper synsets and increase their contributions into the document topics. But the frequencies and contribution of hypernyms into the topic of the document are not changed. And we see the great improvement of the model quality: the kernel uniqueness considerably improves, perplexity decreases to levels comparable with the unigram model, topic coherence characteristics also improve for most collections (Table 2: LDA-Sim+WNsynrel/hyp).

We further use the WordNet-based similarity sets with n-grams having the same components as described in [17]. All measures significantly improve for all collections (Table 2: LDA-Sim+WNsr/hyp+Ngrams). At the last step, we try to apply the same approach to ngrams that was previously utilized to hyponym-hypernym relations: frequencies of shorter ngrams and words are summed to frequencies of longer ngrams but not vice versa. In this case we try to increase the contribution of more specific longer ngrams into topics. It can be seen (Table 2) that the kernel uniqueness grows significantly, at this step it is 1.3–1.6 times greater than for the baseline models achieving 0.76 on the ACL collection (Table 2: LDA-Sim+WNsr/hyp+Ngrams/l).

At the second series of the experiments, we applied EuroVoc information retrieval thesaurus to two European Union collections: Europarl and JRC. In

content, the EuroVoc thesaurus is much smaller than WordNet, it contains terms from economic and political domains and does not include general abstract words. The results are shown in Table 3. It can be seen that inclusion of EuroVoc synsets improves the topic coherence and increases kernel uniqueness (in contrast to results with WordNet). Adding ngrams further improves the topic coherence and kernel uniqueness.

At last we experimented with the Russian banking collection and utilized RuThes thesaurus. In this case we obtained improvement already on RuThes synsets and again adding ngrams further improved topic coherence and kernel uniqueness (Table 4).

Table 4. The results obtained for Russian Banking collection

Collection	Processing	TC-PMI	TC-NPMI	Kernel Uniq	Perplex.
Banking collection	LDA unigram	1.81	0.29	0.54	1654
	LDA+1000ngrams	2.01	0.30	0.60	2497
	LDA-Sim+RTsyn	2.03	0.29	0.63	2189
	LDA-Sim+RTsr/hyp+Ngrams	2.72	0.33	0.70	2396
	LDA-SIM+RTsr/hyp+Ngrams/l	3.02	0.31	0.68	2311

It is worth noting that adding ngrams sometimes worsens the TC-NPMI measure, especially on the JRC collection. This is due to the fact that in these evaluation frameworks, the topics' top elements contain a lot of multiword expressions, which rarely occur in Wikipedia, used for the coherence calculation, therefore the utilized automatic coherence measures can have insufficient evidence for correct estimates.

6 Manual Evaluation of Combined Topic Models

To estimate the quality of topic models in a real task, we chose Islam informational portal “Golos Islama” (Islam Voice)⁴ (in Russian). This portal contains both news articles related to Islam and articles discussing Islam basics. We supposed that the thematic analysis of this specialized site can be significantly improved with domain-specific knowledge described in the thesaurus form. We extracted the site contents using Open Web Spider⁵ and obtained 26,839 pages.

To combine knowledge with a topic model, we used RuThes thesaurus together with the additional block of the Islam thesaurus. The Islam thesaurus contains more than 5 thousand Islam-related terms including single words and expressions.

For each combined model, we ran two experiments with 100 topics and with 200 topics. The generated topics were evaluated by two linguists, who had previously worked on the Islam thesaurus. The evaluation task was formulated as

⁴ <https://golosislama.com/>.

⁵ <https://github.com/shen139/openwebspider/releases>.

follows: the experts should read the top elements of the generated topics and try to formulate labels of these topics. The labels should be different for each topic in the set generated with a specific model. The experts should also assign scores to the topics' labels:

- 2, if the label describes all or almost all elements of ten top elements of the topic
- 1, if the description is partial, that is, several elements do not correspond to the label,
- 0, if the label cannot be formulated.

Then we can sum up all the scores for each model under consideration and compare the total scores in value. Thus, maximum values of the topic score are 200 for a 100-topic model and 400 for a 200-topic model. In this experiment we do not measure inter-annotator agreement for each topic, but try to get expert's general impression.

Table 5. Results of manual labeling of topic models for the Islam site

N	Model	100 topics				200 topics			
		Score	KernU	Prpl	RelC.	Score	KernU	Prpl	RelC.
1	LDA unigram	163	0.535	2520	0.05	334	0.507	2169	0.06
2	LDA+1000phrases	161	0.569	2901	0.06	316	0.534	2494	0.06
3	LDA+ More10phrases	148	0.559	3228	0.05	308	0.527	2774	0.06
4	LDA-Sim+ 1000phrases	180	0.631	2427	0.13	344	0.603	2044	0.11
5	LDA-Sim+ More10phrases	180	0.615	2886	0.14	337	0.596	2398	0.12
6	LDA-Sim+UnarySyn	157	0.632	1999	0.17	323	0.587	1707	0.16
7	LDA-Sim+synrel+ 1000phrases	159	0.622	1797	0.25	301	0.543	1577	0.27
8	LDA-Sim+synrel+ More10phrases	150	0.587	2022	0.26	295	0.526	1758	0.25
9	LDA-Sim+synrel/ hyp+1000phrases	153	0.656	2163	0.26	310	0.603	1900	0.24
10	LDA-Sim+synrel/ hyp+More10phrases	174	0.636	2476	0.24	302	0.244	2476	0.24
11	LDA-Sim+synrel/ GL+More10phrases	186	0.655	1772	0.25	350	0.612	1464	0.25
12	LDA-Sim+synrel/ GL/hyp+ More10phrases	184	0. 686	2203	0.24	346	0.644	1812	0.23

Due to the complicated character of the Islam portal contents for automatic extraction (numerous words and names difficult for Russian morphological analyzers), we did not use automatic extraction of multiword expressions and exploited only phrases described in RuThes or in the Islam Thesaurus. We added thesaurus phrases in two ways: most frequent 1000 phrases (as in [15,17]) and phrases with frequency more than 10 (More10phrases): the number of such phrases is 9351.

The results of the evaluation are shown in Table 5. The table contains the overall expert scores for a topic model (Score), kernel uniqueness as in the previous section (KernU), perplexity (Prpl). Also for each model kernels, we calculated the average number of known relations between topics's elements: thesaurus relations (synonyms and direct relations between concepts) and component-based relations between phrases (Relc).

It can be seen that if we add phrases without accounting component similarity (Runs 2, 3), the quality of topics decreases: the more phrases are added, the more the quality degrades. The human scores also confirm this fact. But if the similarity between phrase components is considered then the quality of topics significantly improves and becomes better than for unigram models (Runs 4, 5). All measures are better. Relational coherence between kernel elements also grows. The number of added phrases is not very essential.

Adding unary synonyms decreases the quality of the models (Run 6) according to human scores. But all other measures behave differently: kernel uniqueness is high, perplexity decreases, relational coherence grows. The problem of this model is in that non-topical, general words are grouped together, reinforce one another but do not look as related to any topic. Adding all thesaurus relations is not very beneficial (Runs 7, 8). If we consider all relations except hyponyms, the human scores are better for corresponding runs (Runs 9, 10). Relational coherence in topics' kernels achieves very high values: the quarter of all elements have some relations between each other, but it does not help to improve topics. The explanation is the same: general words can be grouped together.

At last, we removed General Lexicon concepts from the RuThes data, which are top-level, non-thematic concepts that can be met in arbitrary domains [18] and considered all-relations and without-hyponyms variants (Runs 11, 12). These last variants achieved maximal human scores because they add thematic knowledge and avoid general knowledge, which can distort topics. Kernel uniqueness is also maximal.

Table 6 shows similar topics obtained with the unigram, phrase-enriched (Run 5) and the thesaurus-enriched topic model (Run 12). The Run-5 model adds thesaurus phrases with frequency more than 10 and accounts for the component similarity between phrases. The Run-12 model accounts both component relations and hypernym thesaurus relations. All topics are of high quality, quite understandable. The experts evaluated them with the same high scores.

Phrase-enriched and thesaurus-enriched topics convey the content using both single words and phrases. It can be seen that phrase-enriched topics contain more phrases. Sometimes the phrases can create not very convincing relations such as

Table 6. Comparison of similar topics in the unigram, phrase-based (Run 5) and the best thesaurus-enriched topic models (Run 12).

N	Unigram topic	Phrase-enriched topic	Thesaurus-enriched topic
	Syria topic (Run 1) Relation coherence 0.11	Syria topic (Run 5) Relation coherence 0.13	Syria topic (Run 12) Relation coherence 0.36
1.	сирия (Syria)	сирия (Syria)	сирия (Syria)
2.	сирийский (Syrian)	башар асад (Bashar al-Assad)	сирийский (Syrian)
3.	асад (Assad)	сирийская оппозиция (Syrian opposition)	асад (Assad)
4.	оон (UN)	сирийский (Syrian)	дамаск (Damask)
5.	оппозиция (opposition)	режим асада (al-Assad regime)	башар асад (Bashar al-Assad)
6.	башар (Bashar)	асад (Assad)	сирийская оппозиция (Syrian opposition)
7.	страна (country)	сирийский режим (Syrian regime)	оппозиция (opposition)
8.	дамаск (Damask)	режим башара асада (Bashar al-Assad regime)	режим асада (al-Assad regime)
9.	президент (President)	сирийская власть (Syrian authorities)	режим (regime)
	Orthodox church topic Relation coherence 0.04	Orthodox church topic Relation coherence 0.2	Orthodox church topic Relation coherence 0.33
1.	православный (orthodox)	русская православная церковь (Russian orthodox church)	церковь (church)
2.	церковь (church)	православный (orthodox)	православный (orthodox)
3.	рпц (ROC, abbreviation)	церковь (church)	храм (temple)
4.	патриарх (patriarch)	русский язык (Russian language)	православие (orthodoxy)
5.	храм (temple)	рпц (ROC, abbreviation)	церковный (churchly)
6.	русский (Russian)	православная церковь (orthodox church)	русская православная церковь (Russian orthodox church)
7.	московский (Moscow)	патриарх (patriarch)	духовный (spiritual)
8.	год (year)	кирилл (Kirill)	русский (russian) (Russian)
9.	священник (priest)	государственный язык (state language)	рпц (ROC, abbr. for Russian church)
10.	кирилл (Kirill, orthodox patriarch)	священник (priest)	собор (cathedral)

Russian church - Russian language. It is explainable but does not seem much topical in this case.

The thesaurus topics seem to convey the contents in the most concentrated way. In the Syrian topic general word *country* is absent; instead of *UN* (United Nations), it contains word *rebel*, which is closer to the Syrian situation. In the Orthodox church topic, the unigram variant contains extra word *year*, relations of words *Moscow* and *Kirill* to other words in the topic can be inferred only from the encyclopedic knowledge.

7 Conclusion

In this paper we presented the approach for introducing thesaurus information into topic models. The main idea of the approach is based on the assumption that if related words or phrases co-occur in the same text, their frequencies should be enhanced and this action leads to their mutual larger contribution into topics found in this text.

In the experiments on four English collections, it was shown that the direct implementation of this idea using WordNet synonyms and/or direct relations leads to great degradation of the unigram model. But the correction of initial assumptions and excluding hyponyms from frequencies adding improve the model and makes it much better than the initial model in several measures. Adding ngrams in a similar manner further improves the model.

Introducing information from domain-specific thesaurus EuroVoc led to improving the initial model without the additional assumption, which can be explained by the absence of general abstract words in such information-retrieval thesauri.

We also considered thematic analysis of an Islam Internet site and evaluated the combined topic models manually. We found that the best, understandable topics are obtained by adding domain-specific thesaurus knowledge (domain terms, synonyms, and relations).

Acknowledgments. This study is supported by Russian Scientific Foundation in part concerning the combined approach uniting thesaurus information and probabilistic topic models (project N16-18-02074). The study on application of the approach to content analysis of Islam sites is supported by Russian Foundation for Basic Research (project N 16-29-09606).

References

1. Blei, D.: Probabilistic topic models. Commun. ACM **55**(4), 77–84 (2012)
2. Smith, A., Lee, T.Y., Poursabzi-Sangdeh, F., Boyd-Graber, J., Elmquist, N., Findlater, L.: Evaluating visual representations for topic understanding and their effects on manually generated labels. Trans. Assoc. Comput. Linguist. **5**, 1–15 (2017)

3. Chang, J., Boyd-Graber, J., Wang, Ch., Gerrich S., Blei, D.: Reading tea leaves: how humans interpret topic models. In: Proceedings of the 24th Annual Conference on Neural Information Processing Systems, pp. 288–296 (2009)
4. Boyd-Graber, J., Mimno, D., Newman, D.: Care and feeding of topic models: problems, diagnostics, and improvements. In: Handbooks of Modern Statistical Methods. CRC Press, Boca Raton (2014)
5. Blei, D., Lafferty, J.: Visualizing topics with multi-word expressions (2009). <https://arxiv.org/pdf/0907.1013.pdf>
6. Andrzejewski, D., Zhu, X., Craven, M.: Incorporating domain knowledge into topic modeling via dirichlet forest priors. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 25–32 (2011)
7. Newman, D., Bonilla, E., Buntine, W.: Improving topic coherence with regularized topic models. In: Advances in Neural Information Processing Systems, pp. 496–504 (2011)
8. Xie, P., Yang D., Xing, E.: Incorporating word correlation knowledge into topic modeling. In: Proceedings of NAACL-2015, pp. 725–734 (2015)
9. Chen, Z., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., Ghosh, R.: Discovering coherent topics using general knowledge. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, pp. 209–218. ACM (2013)
10. Mimno, D., Wallach, H., Talley, E., Leenders, M., McCallum, A.: Optimizing semantic coherence in topic models. In: Proceedings of EMNLP 2011, pp. 262–272 (2011)
11. Gao, Y., Wen, D.: Semantic similarity-enhanced topic models for document analysis. In: Chang, M., Li, Y. (eds.) Smart Learning Environments. LNET, pp. 45–56. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-44447-4_3
12. Wallach, H.: Topic modeling: beyond bag-of-words. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 977–984 (2006)
13. Griffiths, T., Steyvers, M., Tenenbaum, J.: Topics in semantic representation. Psychol. Rev. **114**(2), 211–244 (2007)
14. Wang, X., McCallum, A., Wei, X.: Topical n-grams: phrase and topic discovery, with an application to information retrieval. In: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining, pp. 697–702 (2007)
15. Lau, J., Baldwin, T., Newman, D.: On collocations and topic models. ACM Trans. Speech Lang. Process. **10**(3), 1–14 (2013)
16. Nokel, M., Loukachevitch, N.: A method of accounting bigrams in topic models. In: Proceedings of the 11th Workshop on Multiword Expressions (2015)
17. Nokel, M., Loukachevitch, N.: Accounting ngrams and multi-word terms can improve topic models. In: Proceedings of the 11th Workshop on Multiword Expressions (2016)
18. Loukachevitch, N., Dobrov B.: RuThes linguistic ontology vs. Russian wordnets. In: Proceedings of Global WordNet Conference (GWC-2014) (2014)
19. Lau, J., Newman, D., Baldwin, T.: Machine reading tea leaves: automatically evaluating topic coherence and topic model quality. In: Proceedings of the European Chapter of the Association for Computational Linguistics (2014)
20. Bouma, G.: Normalized (pointwise) mutual information in collocation extraction. In: Proceedings of the Biennial GSCL Conference, Potsdam, Germany, pp. 31–40 (2009)
21. Frantzi, K., Ananiadou, S.: The c-value/nc-value domain-independent method for multi-word term extraction. J. Natural Lang. Process. **6**(3), 145–179 (1999)

Russian-Language Question Classification: A New Typology and First Results

Kirill Nikolaev and Alexey Malafeev

National Research University Higher School of Economics,
Nizhny Novgorod, Russia
kinikolaev@edu.hse.ru, aumalafeev@hse.ru

Abstract. This paper deals with automatic classification of questions in the Russian language, a natural early step in building a question answering system. We developed a typology of Russian questions using interrogative particles, pronouns and word order as the main features. A corpus of 2008 questions was manually compiled and annotated according to our typology. We used a fine-grained class set and a coarse-grained one (23 and 14 classes, respectively). The training data, represented as character bi-/trigrams and word uni-/bi-trigrams, was used to approach the task of question classification. We tested several widely used machine-learning methods (logistic regression, support vector machines, naïve Bayes) against a regular expression baseline on a held-out test corpus annotated by an external expert. The best results were achieved by a SVM classifier (linear kernel) that achieved the accuracy of 65.3% (fine-grained) and 68.7% (coarse-grained), while the baseline regular expression model showed 52.7% accuracy.

Keywords: Question answering · Question answering systems · QA systems
Russian-language questions · Question classification · Question tagging
Russian question typology

1 Introduction

Interrogative sentences have a very important function in communication. Questions have a cognitive meaning without a common attributive proposition, because they do not usually include affirmation or negation. They are mostly used for acquiring knowledge and are key tools of cognition (Nevolnikova 2004).

From the natural language processing perspective, interrogative sentences are often key language material when one works on such tasks as question answering, building dialogue agents, and discourse modeling. In this paper, we focus on automatic question typology for the domain of question answering. According to Burger et al. (2001), identifying question classes is the first step in building a QA system.

In contrast to information retrieval (IR) systems, which usually return a set of documents relevant to some keywords, question-answering (QA) systems aim at yielding an exact answer to the question (Monz 2003a, b). Question answering is considered a more difficult task due to the constraints on input (natural language questions vs keywords) and output (focused answers vs entire documents) representation (Bunescu and Huang 2010).

Yet, the benefit of QA systems is that they do not overwhelm the user with excess information (Galea 2003).

Various systems have been developed for answering questions in English (e.g. TEQUESTA (Monz 2003a, b), START (Katz et al. 2006), OpenEphyra (van Zaanen 2008), IBM Watson (Ferucci et al. 2010), EAGLi (Gobelli et al. 2012), etc.). An excellent survey of state-of-the-art learning-based methods for English-language question classification was conducted by Loni (2011).

As far as Russian-language QA systems are concerned, they have had very limited coverage in literature. Although a monograph by Sosnin (2007) and a few research papers, such as (Suleymanov 2001; Tikhomirov 2006; Mozgovoy 2006; Solov'ev and Peskova 2010), have been published, they mostly contribute to the theory on the problem, rather than propose and/or evaluate efficient practical solutions.

Our work differs from the papers previously published by other researchers in that we attempt to solve the practical task of automatic typologization of questions in the Russian language using complex class sets (23 question types in the fine-grained and 14 in the coarse-grained set), while in other papers other approaches have been used. For example, in (Sosnin 2007), only three general question classes are distinguished: (1) “problem”, (2) “task”, and (3) “inquiry” (p. 118). Alternatively, in (Solov'ev and Peskova 2010), a paper devoted to question analysis for a Russian-language question answering system, a detailed question taxonomy was used. It was borrowed from (Ittycheriah 2008) with some modification, but we consider it too fine-grained to be practical. It has 39 classes, some of which are very specific and rare, e.g. *Organ*, *Salutation*, *Plant*, etc. Also, the differences between some of the classes do not seem very well-defined, e.g. *Areas* vs *Geological objects* vs *Location* vs *Country*, or *Company-roles* vs *Occupation*, etc. Furthermore, the question analysis technique used was trivial: a limited number of key words were simply searched for in the question. Expectedly, the performance of the module was quite low – 67% error (Solov'ev and Peskova 2010, p. 48). In contrast, we report question tagging accuracy of up to 68.7%. That said, the results are obviously not directly comparable because different classifications are used.

In the next section, we describe the question typology that the developed classifier is based on. Sections 3 and 4 are devoted to the regular expression baseline and the machine learning classification methods used, respectively. Section 5 discusses the results attained. Finally, in Sect. 6 we draw conclusions and outline some directions for future work.

2 Interrogative Sentences in the Russian Language

In our research, we focus on the functional aspect of interrogative sentences. According to Shvedova, information gathered via interrogative sentences can be of various nature: about the subject of some action (*Кто это сделал?/Who did it?*), the object (*Что было сделано?/What was done?*), the goal (*Для чего ей это?/Why does she need this?*), etc. (Shvedova 1980). The main question formation means are intonation, interrogative particles (*ли*, *так*, *верно*, *как*, *что ли*, *etc.*), interrogative pronominal

words (*где, куда, когда, откуда, почему, зачем, как, etc.*) and word order. It is possible to use some of these means as features in classification.

The first question typology that we considered was developed by Shvedova (1980). She divides interrogative sentences as follows:

1. According to the type and volume of required information:
 - a. General: acquiring the information about the situation in whole. *Что происходит в Китае?/What is happening in China?*
 - b. Special: acquiring the information about an aspect of the matter. *Какие песни сделали её известной?/Which songs made her famous?*
2. According to what answer is expected:
 - a. Requiring a confirmation: yes/no or true/false: *Лондон – столица Англии?/Is London the capital of England?*
 - b. Requiring information: *Зачем они пьют чай?/What do they drink tea for?*

It can be seen that this typology is too general for the purposes of automatic question answering. Another functional classification we used for building our own typology is Graesser's Taxonomy of Inquiries (Lauer et al. 2013). The taxonomy is presented in Table 1.

Table 1. Arthur Graesser's Taxonomy of Inquiries

Question	Abstract specification
Verification	Is a fact true?
	Did an event occur?
Comparison	How is X similar to /different from Y?
Disjunctive	Is X or Y the case?
Concept completion	Who? What? When? Where?
Definition	What does X mean?
Example	What is an example of X?
Interpretation	How is the particular event interpreted or summarized?
Feature specification	What qualitative attributes does X have?
Quantification	What is the value of a quantitative variable?
Casual antecedent	What caused some event to occur?
Casual consequence	What are the consequences of an event/state?
Goal orientation	What are the motives behind an agent's actions?
Enablement	What object or resources enable an agent to perform an action?
Instrumental	How does an agent accomplish the goal?
Expectational	Why did some expected event not occur?
Judgmental	The questioner wants the answerer to judge an idea or give an advice what to do
Assertion	The speaker expresses that he or she is missing some information
Request/Directive	The speaker directly requests the information

Based on these classifications and drawing upon the functional aspect of interrogative sentences in Russian, we created a new question typology for the purposes of our research (Table 2).

Table 2. Russian question typology

Tag	Numeric tag	Wording examples
General	1	<i>Что происходит в ...?/What is happening in ...?</i>
Verification	2	<i>Правда ли, что ...?/Is it true that ...?</i>
Definition	3	<i>Что означает/такое?/What is ...? What does ... mean?</i>
Example	4	<i>Приведи пример...?/Give an example of ...?</i>
Comparison	5	<i>Чем похожи/отличаются...?/What are the similarities/differences between ...?</i>
Choice	6	<i>X или Y?/X or Y?</i>
Concept Completion	7	[Further subdivided into:]
(a) Agent	71	<i>Кто?/Who?</i>
(b) Object	72	<i>Что?/What?</i>
(c) Location	73	<i>Где? Куда? Откуда?/Where? From where?</i>
(d) Date:	74	<i>Когда? Какого числа?/When? What date?</i>
• Date-birth	741	<i>Когда родился?/When was ... born?</i>
• Date-death	742	<i>Когда умер?/When did ... die?</i>
(e) Time	75	<i>Во сколько?/What time?</i>
Quality	8	<i>Какой?/What kind of?</i>
Quantity	9	<i>Сколько? Как много?/How many/much?</i>
(a) Quantity-age	91	<i>Сколько лет? В каком возрасте?/How old? At what age?</i>
(b) Quantity-time	92	<i>Сколько времени?/What time?</i>
(c) Price	93	<i>Сколько стоит?/How much is ...?</i>
Action	10	<i>Что делать, чтобы...?/What do I do to/if ...?</i>
Instrument	11	<i>С помощью чего? Каким методом?/With what? How?</i>
Goal	12	<i>К чему? Зачем?/For what?</i>
Reason	13	<i>Почему?/Why?</i>
Consequence	14	<i>Каковы последствия?/What are the consequences of ...?</i>

As can be seen from the Table, there are 23 distinct question classes in our typology. Since some classes are further subclassed, this is also a taxonomy of question types. If we flatten the taxonomy, we get 14 general classes (the coarse-grained class set). We use both fine-grained and coarse-grained class sets in our experiments (Sect. 4).

Some comments need to be made on our classification. Firstly, the proposed classification is by no means a complete listing of all possible question types in Russian. When developing the classification, we aimed at the balance between completeness and practicality. Thus, certain question types not so commonly used in

practice were omitted: e.g. *как часто?* (*how often?*) or *до какого времени?* (*up to what time?*).

Another important comment is that not all wordings for each question type are listed in the table; for the sake of brevity, we list only the most common ones, although other wordings are possible. For example, apart from *почему?* (*why?*), a question of the “Reason” type may be worded as: *по какой причине?* (*for what reason?*), *отчего?* (*why?*), etc.

The “General” question type might need explanation. In this type of questions, a general summary of a situation is requested, e.g. *Что происходит в Китае?*/*What is happening in China?* In contrast, “Verification” questions require a short yes/no answer, e.g. *Верно ли, что кошки не летают?*/*Is it true that cats don't fly?*

In the following two sections, we will describe our attempt at solving the Russian question classification task using the developed typology of questions.

3 Baseline Question Tagging Method: Regular Expressions

Most QA systems first classify questions based on the type (related to such question words as “What”, “Why”, “Who”, “How”, “Where”), which is followed by the identification of the answer type (Damljanovic et al. 2010). Manually designed regular expression are the most obvious tool for identifying the question type, and they are successfully used in many QA systems, e.g. TEQUESTA (Monz 2003a, b).

We implemented our own regular expressions-based question classifier in Python 3. With a more or less complex pattern for each question type, this classifier was used as the baseline method in our work. Some examples of the patterns are given in Table 3.

Table 3. Sample regular expression patterns for Russian interrogative sentences

Tag	Numerical tag	Example patterns
Example	4	/.*(([пП]ривели [кК]акой пример образец) (([чЧ]то [кК]то) ((может ((служить) (выступ(ать ить)))) ((по?)служит выступ(ает ит))) (как)? ((пример(ом?)) (образ(ец цом))))).*[?!.]/
Quality	8	/.*(([кК]ак(ой ая ое ом ие)) ([кК]ак(им ими ие) ((свойств(ом ами а)) (качеств(ом ами а))) (наделен обладает характеризуется отличается имеет)).*[?!.]/
Goal	12	/.*(([кК] чему) ([зЗ]ачем) ([сС] как(ой ими) (цел(ью ями)) ([задач(ей ами)) ([дД]ля чего) (([вВ]о имя) ([дД]ля как(ой их)) (цел(и ей)) (задач(и?)) ([кК]ак(ую ие) (цел(ы и)) ([задач(у и))).*[?!.]/

In our implementation, identifying the type of the question with regular expressions is incremental. A string variable is gradually matched, via `re.match(pattern, string)` method, against all example patterns, starting with the simplest (1, 2, 71, 72, 73), and ending with the most complex (8, 10, 11). The latest matching pattern is chosen as the hypothetic answer.

We evaluated the baseline classifier (as well as other classifiers described in the next section) on a held-out test set of 150 questions manually annotated by an external expert in accordance with our classification. The regular expression classifier correctly matched 79 questions (52.7% accuracy).

4 Applying Machine Learning to Question Tagging in Russian

Many early approaches to determining question types employ manually designed rules or regular expressions and are non-probabilistic in that a pattern/set of conditions is either matched or not. Pinchak and Lin point out in (2006), however, that such an approach has two major drawbacks:

1. There will always be questions whose types do not match the patterns;
2. The predetermined granularity of the categories leads to a trade-off between how well they match the actual question types and how easy it is to build taggers and classifiers for them.

Thus, a probabilistic answer type model that directly computes the degree of match between a potential answer and the question context is much more effective. Such algorithms are used in some works on QA for English: (Li and Rot 2002; Zhang and Le 2003; Pereira et al. 2009), etc.

To train our question classifiers, we needed a set of questions in Russian, annotated for question types. Due to the unavailability of a suitable collection of questions, we semi-automatically extracted a set of 2008 questions from the Russian Internet Corpus (Sharoff et al. 2006), and manually tagged each question in accordance with our typology. We used two different annotation sets: with the simplified “Concept Completion” and “Quantity” class groups (14 classes total) and the original detailed classification (23 classes).

The questions were converted to five different bag-of-word representations each (character bigrams, character trigrams, word unigrams, word bigrams, word trigrams) and then used for the learning of some traditional machine learning algorithms. This was done in RapidMiner, a cross-platform software framework developed on an open-core model and providing multiple solutions for machine learning, data and text mining, predictive analytics, etc. (Klinkenberg 2013).

For automatic classification of questions, we trained (using the ‘1 vs All’ classification strategy) three different machine learning algorithms - naïve Bayes, support vector machine and logistic regression. Ten different datasets, depending on the set of classes – fine-grained or coarse grained, – and type of representation (word unigrams/bigrams/trigrams, and character bigrams/trigrams) were used with each algorithm. During evaluation, we ensured that the system relied only on the n-grams observed in training. All n-grams not seen in the training data were ignored, i.e. no strategy for dealing with unknown (not previously seen) n-grams was implemented.

The trained models were tested on a held-out test set of 150 questions manually annotated by an external expert in accordance with our classification. The training and test sets of questions, as well as the models used in this study, are made freely available

to the community¹. The distribution of coarse-grained question types in the training and test sets is illustrated in Table 4. Table 5 shows the evaluation results for both the fine-grained and coarse-grained class sets.

Table 4. Question type distribution in the training and test sets.

Set	1	2	3	4	5	6	7	Total
Training	153	424	64	10	15	76	579	
Test	15	34	6	3	2	10	25	
	8	9	10	11	12	13	14	
Training	186	49	59	9	116	201	67	2008
Test	17	10	1	2	5	16	4	150

Table 5. Prediction accuracy for question tagging.

Algorithm and model	Correct matches (fine-grained)	Accuracy (fine-grained)	Correct matches (coarse-grained)	Accuracy (coarse-grained)
NB, c-bi	23	15.3%	19	12.7%
NB, c-tri	39	26%	39	26%
NB, w-uni	38	25.3%	39	26%
NB, w-bi	61	40.7%	61	40.7%
NB, w-tri	61	40.7%	62	41.3%
SVM, c-bi	65	43.3%	76	50.7%
SVM, c-tri	69	46%	73	48.7%
SVM, w-uni	67	44.7%	74	49.3%
SVM, w-bi	73	48.7%	82	54.7%
SVM, w-tri	68	45.3%	81	54%
LR, c-bi	55	36.7%	63	42%
LR, c-tri	20	13.3%	40	26.7%
LR, w-uni	58	38.7%	60	40%
LR, w-bi	90	59.3%	92	61.3%
LR, w-tri	88	56%	97	64.7%

We also studied how prediction quality changed with the normalization of the training data set and using different SVM kernels. This is illustrated in Table 6:

Thus, the best classification result (65.3% acc. fine-grained and 68.7% acc. coarse-grained) was achieved by a support vector machine with a linear kernel, using the word trigram text representation. This is consistent with the state-of-the-art results for English question classification reported in (Silva et al. 2011), also obtained with a linear SVM classifier.

¹ <https://github.com/Pythonimous/Q-A-System>.

Table 6. Normalization, linear SVM kernel and system performance

Algorithm, setup	Correct matches (fine-grained)	Accuracy	Correct matches (coarse-grained)	Accuracy
SVM (dot kernel), w-tri, normalized proportion	68	45.3%	81	54%
SVM (linear kernel), w-tri, normalized proportion	98	65.3%	103	68.7%

5 Discussion

After evaluating all classification algorithms trained on the different datasets, we can observe the following. Firstly, as expected, using the coarse-grained class set was an easier task that resulted in higher accuracy than for the fine-grained set. The accuracy difference ranged from 0.6% for naïve Bayes to 8.7% for logistic regression (considering the best text representation models for each ML algorithm). Using normalization and the linear kernel for the SVM allowed us to increase classification accuracy to its highest of 65.3% and 68.7% for the fine-grained and the coarse-grained class sets, respectively. To compare, the regular expression baseline showed a stable 52.7% accuracy, which proved considerably better than naïve Bayes, but less effective than the two other algorithms.

The accuracy is quite low in comparison with the results obtained for English-language datasets. Indeed, the state-of-the-art result reported by Silva et al. (2011) is 95% and 90.8% for 6 coarse-grained and 50 fine-grained classes, respectively. However, this is to be expected for a number of reasons. Firstly, Silva et al. used a much larger dataset, published by Li and Roth (2002), consisting of 5500 training and 500 test questions. Secondly, there are many freely available NLP tools for English, which allows for extracting various features useful for question classification. In particular, apart from word unigrams, the classifier by Silva et al. used such features as headwords, hypernyms and indirect hypernyms. Lastly, a lot of research was published on question classification for English, as shown in (Loni 2011), which cannot be said about Russian-language question tagging. Since the Russian language is very different from English in both morphology and syntax, the methods used for the classification of questions in English are not always equally effective or even applicable for Russian.

The confusion matrix (not presented here due to size constraints, but available upon request) for the top-accuracy algorithm was also analyzed. It was found that the “Instrument”, “Example”, “Action”, and “Definition” questions were predicted with less than 50% accuracy. These categories were present in training set with a low frequency: 0.004, 0.004, 0.029, and 0.031 per 1000 questions, respectively. The most accurately predicted question types were “Consequence”, “Reason” and “Goal”.

6 Conclusion and Future Work

The main theoretic contribution of our paper is a classification of Russian questions, consisting of 14 coarse-grained and 23 fine-grained classes. Using this typology, we have applied machine learning methods to the task of automatic classification of

Russian questions. We tested a regular expression baseline and three different classifiers using five different sets of features (character bi-/trigrams, word uni-/bi-/trigrams). The best classifier, SVM (linear kernel), trained on pre-normalized (via proportion transformation) word trigrams, achieved the classification accuracy of 65.3% and 68.7% for the fine-grained and the coarse-grained class sets, respectively, in contrast to the 52.7% baseline result (regular expression model).

Presented in this paper is one of the very few attempts to solve the Russian question tagging task using large class sets. The methods employed in our work showed relatively good results (given the class set sizes) in comparison with what is reported for Russian-language question classification in the literature, although there is still a lot of room for improvement. Indeed, much work needs to be done to approach the above 90% accuracy achieved for English by the research community.

Our work can be used for building a complete Russian QA pipeline or as a standalone question tagging solution. We believe that it should be possible to improve our results by further expanding the training data set, as well as using more complex machine learning algorithms (neural networks) and features (word2vec).

References

- Bunescu, R., Huang, Y.: Towards a general model of answer typing: question focus identification. In: Proceedings of the 11th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2010), RCS Volume, pp. 231–242 (2010)
- Burger, J., Cardie C., Chaudhri V., Gaizauskas R., Harabagiu S., Israel D., Jacquemin, C., Lin, C. Y., Maiorano, S., Miller, G., Moldovan, D.: Issues, tasks and program structures to roadmap research in question & answering (Q&A). In: Document Understanding Conferences Roadmapping Documents, pp. 1–35 (2001)
- Damljanovic, D., Agatonovic, M., Cunningham, H.: Identification of the Question focus: combining syntactic analysis and ontology-based lookup through the user interaction. In: LREC (2010)
- Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A.A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J., Schlaefer, N.: Building Watson: an overview of the DeepQA project. *AI Mag.* **31**(3), 59–79 (2010)
- Galea, A.: Open-domain surface-based question answering system. In: Proceedings of CSAW, vol. 3 (2003)
- Gobeill, J., Pasche, E., Teodoro, D., Veuthey, A.L., Ruch, P.: Answering gene ontology terms to proteomics questions by supervised macro reading in Medline. *EMBnet Journal* **18**(B), 29–31 (2012)
- Ittycheriah, A.: A statistical approach for open domain question answering. In: Strzalkowski, T., Harabagiu, S.M. (eds.) *Advances in Open Domain Question Answering*, vol. 32, pp. 35–69. Springer, Dordrecht (2008). https://doi.org/10.1007/978-1-4020-4746-6_2
- Katz, B., Borchardt, G.C., Felshin, S.: Natural language annotations for question answering. In: FLAIRS Conference, pp. 303–306 (2006)
- Klinkenberg, R. (ed.): *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. Chapman and Hall/CRC, Boca Raton (2013)
- Lauer, T.W., Peacock, E., Graesser, A.C.: *Questions and Information Systems*. Psychology Press, Routledge (2013)

- Li, X., Roth, D.: Learning question classifiers. In: Proceedings of the 19th International Conference on Computational Linguistics, Association for Computational Linguistics, vol. 1, pp. 1–7 (2002)
- Loni, B.: A survey of state-of-the-art methods on question classification. Literature survey, Published on TU Delft Repository (2011)
- Monz, C.: Document retrieval in the context of question answering. In: Sebastiani, F. (ed.) Advances in Information Retrieval, ECIR 2003. LNCS, vol. 2633, pp. 571–579. Springer, Heidelberg (2003a). https://doi.org/10.1007/3-540-36618-0_44
- Monz, C.: From Document Retrieval to Question Answering. Institute for Logic, Language and Computation (2003b)
- Mozgovoy, M.V.: A simple question-answering system based on a semantic analyzer for the Russian language [Prostaya voprosno-otvetnaya sistema na osnove semanticheskogo analizatora russkogo yazyka], Vestnik of the St. Petersburg University. Series 10. Applied mathematics. Informatics. Management processes [Vestnik SPbGU. Seriya 10. Prikladnaya matematika. Informatika. Protsessy upravleniya], no. 1, pp. 116–122 (2006)
- Nevolnikova, S.V.: Functional and semantic types of Russian interrogative sentences and their role in text formation [Funktional'no-semanticheskie raznovidnosti russkikh voprositel'nykh predlozheniy i ikh rol' v tekstoobrazovaniy]. Rostov-on-Don (2004)
- Pereira, F., Mitchell, T., Botvinick, M.: Machine learning classifiers and fMRI: a tutorial overview. *Neuroimage* **45**(1), S199–S209 (2009)
- Pinchak, C., Lin, D.A.: Probabilistic Answer Type Model. In: EACL (2006)
- Sharoff, S.: Creating general-purpose corpora using automated search engine queries. In: WaCky, pp. 63–98 (2006)
- Shvedova, N.Y.: Russkaja Grammatika [Russian Grammar]. AN SSSR Publ, Moscow (1980)
- Silva, J., Coheur, L., Mendes, A.C., Wichert, A.: From symbolic to sub-symbolic information in question classification. *Artif. Intell. Rev.* **35**(2), 137–154 (2011)
- Solov'ev, A.A., Peskova, O.V.: Building a question-answering system for the Russian language: question analysis module [Postroenie voprosno-otvetnoy sistemy dlya russkogo yazyka: modul' analiza voprosov], New information technologies in automated systems [Novye informatsionnye tekhnologii v avtomatizirovannykh sistemakh], no. 13, pp. 41–49 (2010)
- Sosnin, P.I.: Question-Answer Modeling in the Development of Automated Systems [Voprosno-otvetnoe modelirovanie v razrabotke avtomatizovannykh sistem]. Ul'yanovsk, USTU (2007)
- Suleymanov, D.S.: A study of the basic principles of building a semantic interpreter for questions and answers in natural language in AOS [Issledovanie bazovykh printsimov postroeniya semanticheskogo interpretatora voprosno-otvetnykh tekstov na estestvennom yazyke v AOS]. Educational technologies and society [Obrazovatel'nye tekhnologii i obshchestvo], no. 3, pp. 178–192 (2001)
- Tikhomirov, I.A.: Question-answering search in the intelligent search system Exactus [Voprosno-otvetnyy poisk v intellektual'noy poiskovoy sisteme Exactus]. In: Proceedings of the Fourth Russian Seminar on Evaluation of Information Retrieval Methods ROMIP [Trudy chetvertogo rossiyskogo seminara po otsenke metodov informatsionnogo poiska ROMIP], pp. 80–85 (2006)
- van Zaanen, M.: Multi-lingual Question Answering using OpenEphyra. CLEF (Working Notes) (2008)
- Zhang, D., Lee, W.S.: Question classification using support vector machines. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 26–32 (2003)

Domain Adaptation for Resume Classification Using Convolutional Neural Networks

Luiza Sayfullina^(✉), Eric Malmi, Yiping Liao, and Alexander Jung

Department of Computer Science, Aalto University, Espoo, Finland
sayfullina.luiza@gmail.com

Abstract. We propose a novel method for classifying resume data of job applicants into 27 different job categories using convolutional neural networks. Since resume data is costly and hard to obtain due to its sensitive nature, we use domain adaptation. In particular, we train a classifier on a large number of freely available job description snippets and then use it to classify resume data. We empirically verify a reasonable classification performance of our approach despite having only a small amount of labeled resume data available.

Keywords: Resume classification · Convolutional neural networks
Job-market analysis

1 Introduction

The fast paced online job-market industry requires recruiters to screen through vast amounts of resume data in order to evaluate applicants fast and reliable. The design of accurate automatic classification systems typically requires the availability of labeled resume data which can be used to train the classifier.

Due to its sensitivity, resume data is difficult and costly to obtain. In contrast, data about job descriptions can be obtained much easier. However, the two domains constituted by resume data and job description data are intrinsically related. Indeed, both data domains are related to the same job recommendation task, which is to match applicant to suitable job offers. Moreover, the resumes of applications have semantic similarities with job descriptions which belong to the same job category. For instance, they both can contain skills, education, duties as well as personal characteristics of the desired candidate.

So far, there are two main flavors along with their hybrids [9], of job recommendation systems. One class, referred to as content-based recommendation systems, is based mainly on the available job descriptions. A second class, referred to as collaborative filtering recommendation systems, is mainly based on the preferences of users who are interested in similar jobs. Content-Based recommendation system suggests to a user textually similar jobs to what he/she viewed or liked previously [2].

It seems therefore reasonable to use transfer learning in order to implement a domain adaptation in order to leverage the information contained in

vast amounts of labeled job description data in order to classify resume data. Since resume and job summaries belong to similar domains, we expect features extracted by a convolutional neural network for job classification to be highly relevant for resume summaries as well.

The theory of learning in different domains was theoretically approached in [3, 4]. The authors provided generalization bound for domain adaptation using \mathcal{H} -divergence. It consists of two components and tries to find a trade-off between source-target similarity and source training error. Based on that assumption several researchers [1, 6, 7] came up with the domain-adversarial approach, where high-level representations from neural network are optimized to minimize the loss on the source domain and maximize the loss on the domain classifier. [13] proposed another approach based on convolutional neural network special architecture. First three layers are domain-invariant, next two layers are fine-tuned and fully connected layers aim to fit specific tasks, but regularized by multiple kernel variant of maximum mean discrepancy that enforces distributions to be similar. The proposed network is optimized for the image domain, being more specific. In the natural language processing domain adaptation is applied, e.g. for sentiment analysis [8] and phrase extraction from the user reviews [16].

Convolutional neural networks (CNN) have been successfully applied to not only image, but also text classification [12, 17], provided that enough training data is available. We propose a domain adaptation approach [5, 8] where we train a CNN based classifier on 85,000 job description snippets which are labeled using 27 industrial job classes. After the classifier has been trained, we apply it to classify unlabeled resume data.

The paper is organized as follows. First, in Sect. 2 we describe job, resume and children dream job datasets, used for classification. Then in Sect. 3, we describe the `fastText` baseline model and the CNN for short text classification model. Experimental results are provided in Sect. 4, where classification accuracies are reported along with t-SNE visualization built on latent CNN representations. Finally, we present conclusions in Sect. 5.

2 Datasets

We study three different datasets: *job descriptions*, which are used for training models, *resume summaries*, which are our main target domain used for testing the models. *Children’s dream job* descriptions is rather a toy data lacking enough samples for fair evaluation, but these job descriptions significantly differ and thus are interesting to experiment with.

2.1 Job Descriptions

We collected 90,000 job description snippets using the Indeed Job Search API¹, that enables access to short job summaries given a key word. As key words, we used 27 different industrial job categories listed in Table 1.

¹ <https://www.indeed.com/publisher>.

Here is an example of a job summary from the category *Accountant*:

“Entering journal entries, posting cash, and account reconciliations/supporting schedules. This position is responsible for supporting the daily operations and ...”.

Note that the snippets provided by Indeed are generated based on the full description of the job postings, thereby they encapsulate only the condensed information regarding the job. Furthermore, since the descriptions are unstructured text snippets, the contents provided by different companies for similar positions may be inconsistent. For example, some job snippets or summaries do not include informative sentences or keywords related to job titles or categories. However, since the descriptions are not limited by a predefined structure, they may provide richer and more detailed information about the jobs in varying industries.

Table 1. 27 industrial job categories from <https://www.indeed.com/find-jobs.jsp>.

1. Accounting/Finance	10. Banking/Loans	19. Education/Training
2. Healthcare	11. Human Resources	20. Legal
3. Non-Profit/ Volunteering	12. Restaurant/Food service	21. Telecommunications
4. Administrative	13. Construction/Facilities	22. Engineering/ Architecture
5. Computer/ Internet	14. Insurance	23. Manufacturing/ Mechanical
6. Pharmaceutical/ Bio-tech	15. Retail	24. Transportation/ Logistics
7. Arts/Entertainment/ Publishing	16. Customer Service	25. Government/Military
8. Hospitality/ Travel	17. Law Enforcement/Security	26. Marketing/ Advertising/PR
9. Real Estate	18. Sales	27. Upper Management/ Consulting

2.2 Resume Summaries

We collected 523 anonymous resume data samples, each sample labeled with one of the 27 categories based on the type of a job the candidate is looking for. The distribution of the categories is shown in Fig. 1.

Here is an example of a resume self-description summary:

“Experienced analyst with an excellent academic profile and having several years of invaluable experience in domestic and international consultancy

and management. Highly focused with a comprehensive knowledge and understanding of project management, technical issues and financial practices. Good at meeting the deadlines. Consider myself to be sociable person and good team worker.”

2.3 Children’s Dream Jobs

Children, unlike grown-ups, can express their dream jobs more emotionally, without being attached to skills, but rather following their interests. So in addition to resumes, we decided to use children dream job descriptions that were categorized manually into the same 27 job categories. The data set contains 98 children’s short essays on their dream job parsed from². Below is an example essay:

“As far as I can remember I have always wanted to become a medical doctor. More specifically, a cardiologist. I love the thought of saving a person’s life. The road to becoming a doctor is a long process, but worth it in the end. Having the feeling of accomplishment and knowing that I have made an impact on a family’s life, would be the greatest satisfaction for me.”

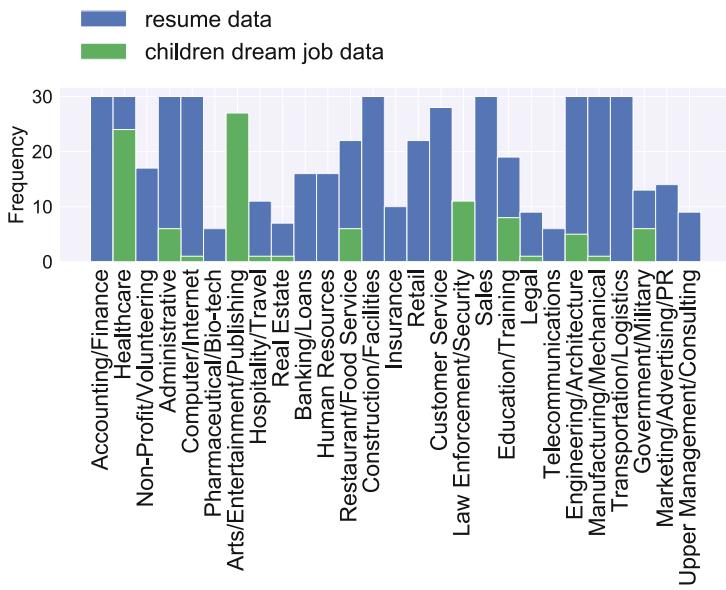


Fig. 1. The comparison of class distribution in job description and resume datasets.

² http://www.valleymorningstar.com/sie/what_do_you_think/article_692e1ac9-bae5-5705-8005-c22dac04ebf6.html.

2.4 Comparison of Job Descriptions and Resumes

Since our aim is to leverage easily available job description data to train a model for classifying resume summary snippets, it is important to understand how these two domains differ from each other. In order to compare the two, we study word frequencies to see whether certain terms are over-represented in one domain compared to the other.

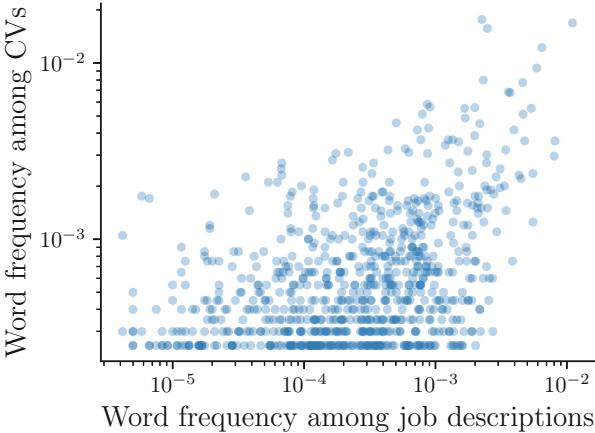


Fig. 2. Normalized frequencies of all the words appearing at least five times in both datasets. Each word corresponds to a dot whose x and y coordinates denote the frequencies among job descriptions and CVs (resumes), respectively.

Figure 2 shows the normalized frequencies of all words appearing at least five times in both datasets. The two frequencies are correlated ($\rho = 0.59$), but we can see, that for some words, the frequencies differ considerably. In Table 2, we list the words for which the relative difference is the largest.³ The results show that in resumes people are much more likely to use adjectives describing themselves, such as *adaptable* and *polite*, whereas job descriptions mention more often roles, such as *director* and *coordinator*.

3 Industrial Category Classification Methods

The objective of industrial category classification is to classify user profiles, represented as text snippets, into 27 industrial categories shown in Table 1. We apply a CNN based methods to this task because they have shown state-of-art performance in text classification [11]. As a baseline method, we employ the `fastText` classifier [10] which is presented next.

³ The relative difference is measured by dividing the two normalized frequencies. For low frequencies, this measure will be noisy but we ignore this since the purpose of the experiment is to merely gain an overview of the differences between the datasets.

Table 2. Words which normalized frequency differs the most between job descriptions (f_{Job}) and resume summaries (f_{CV}). Difference is measured by dividing the frequencies.

	f_{CV}/f_{Job}	Word	f_{Job}/f_{CV}	Word
1	284.9	uk	15.3	program
2	242.2	gained	10.4	assist
3	239.3	adaptable	9.0	director
4	95.0	polite	8.4	medical
5	82.1	keen	6.7	provides
6	76.0	bsc	6.7	coordinator
7	73.3	trustworthy	6.6	accounting
8	73.3	ambition	6.5	executive
9	63.3	licence	6.2	representative
10	59.6	confident	5.9	assistant
11	59.5	adapt	5.8	report
12	57.0	versatile	5.7	food
13	57.0	consultancy	5.7	perform
14	52.9	approachable	5.7	equipment
15	48.8	punctuality	5.6	related

3.1 Fast Text Classifier

The **fastText** method has been proposed recently by Joulin et al. [10] to efficiently classify text data. The method is based on learning word embeddings, averages them, and feeds the resulting vector into a linear classifier. The method also supports learning word embeddings for n-grams which allows capturing word order information.

Supported by a few algorithmic and implementation improvements, **fastText** is able to train and test extremely fast without access to GPUs. We have chosen fastText, since it was shown by Joulin et al. [10] to be a competitive baseline for deep learning models, outperforming models like CNN, char-CNN and slightly (1%) underperforming LSTM-GRNN models.

3.2 Convolutional Neural Networks for Sentence Classification

Word2vec model [15] is a widely used method for learning vector representations of words, so that semantically similar words are close to each other in the vector space. Based on the word vectors, contextual information can be extracted to learn the semantic similarity between words and sentences.

Convolutional neural networks trained on the top of pre-trained word2vec representations proposed by [11] showed state-of-the-art performance on several datasets, including sentiment analysis. In this model, words in the sentences are

embedded word2vec representation of the same length. Then vectors of words are concatenated by rows thus forming a matrix, to which CNN is applied.

Let us introduce some notations. First, $\mathbf{x}_i \in \mathbb{R}^k$ is the i -th word embedded into a vector of k dimensions. A sentence which consists of n words is represented as $\mathbf{x}_{1:n} = \mathbf{x}_1 \oplus \mathbf{x}_2 \dots \oplus \mathbf{x}_n$, where \oplus is the concatenation operator. The convolutional filter with window size of h words is denoted as $\mathbf{w} \in \mathbb{R}^{hk}$. Then the new feature map is generated by:

$$c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b), \quad (1)$$

where $b \in \mathbb{R}$ is a bias term and f is a hyperbolic tangent. Following the convolution operation, the max-pooling operation is applied to capture the most important feature and the output is forwarded to a fully connected soft-max layer whose output is a probability distribution over classes. The regularization of the network is done by applying dropout to prevent co-adaptation and re-scaling weights to prevent large (and possibly noisy) gradient updates during training. At the testing phase, the learned weight vectors are scaled by $\mathbf{w} \leftarrow p\mathbf{w}$. Additionally, an L2-norm constraint is applied to rescale \mathbf{w} to have $\|\mathbf{w}\|_2 = s$ when $\|\mathbf{w}\|_2 > s$ after gradient decent step.

4 Experimental Results

We trained our models by fixing training (80,000 samples) and validation data (5,000 samples) consisting of job summaries and used all available samples from resume and children’s dream job data for testing. We also used 5,000 job summary samples for testing a classifier on purely job data. All selected data samples were trimmed to 100 words.

Our CNN model was based on the implementation by Kim⁴. In order to avoid strong overfitting, we increased the L2-norm constant up to 10 and set the width of CNN filters to be [2–4] instead of [2–5]. We tried the filters of size [50, 100, 200] per each type and have chosen 50 based on validation set from job description data. The dropout rate was set to 0.5 and we found it useful for model regularization. A non-static setting of CNN was chosen, where Google-News pretrained word vectors are fine-tuned while training.

For the **fastText** model, we optimized the lengths of the n-grams and the learning rate hyperparameters using the validation data, obtaining the values 4 and 0.25, respectively. These values were kept fixed for all three test datasets.

The overall prediction accuracies are shown in Table 3. When moving from the source domain to the target domain, the accuracy drops from 74.88% to 40.15%. CNN outperforms **fastText** for each dataset and particularly for resume and dream job data, which shows that the CNN model generalizes better to new domains.

The confusion matrices for job description and resume summary classification are shown in Fig. 3.

⁴ https://github.com/yoonkim/CNN_sentence.

Table 3. Job category prediction accuracies (%) for the fastText method and CNN for short text classification.

Dataset	fastText	CNN
Job description	71.99	74.88
Resume	33.40	40.15
Children’s dream job	28.5	51.02

From Fig. 3 we can see that the hardest categories to classify in job description dataset are Management, Administrative, Sales, Customer Service and Manufacturing. Probably, it happens due to semantic closeness of some job categories, like Management and Administrative, Sales and Retail, since even humans can have trouble clearly distinguishing between them. Manufacturing category samples were classified with Construction, Engineering and Transportation/Logistics labels. The highest recall belongs to Legal, Real Estate, Arts, Law and Non-Profit categories.

Resume dataset has a small number of samples per class, so we can not make general conclusions from confusion matrix showed in Fig. 3. Still, the results on our resume data have common trends with job data. For example, similarly to job confusion matrix, Management, Administrative, Customer Service, Retail and Manufacturing categories have a low recall. Legal, Government, Arts, Healthcare and Pharmacy show the highest recall. Management category, consisting of 9 samples, was not detected at all, probably since this position can be quite general and related to various job fields.

One of the ways to achieve better generalization is building latent representations. In our case, the concatenated outputs c_i of the first layer of the CNN model form latent space representations. Therefore, we visualized those outputs both for job and resume test data using t-SNE [14] projection and show the results in Fig. 4.

One can observe the presence of category clusters formed by job samples, although some of them are not perfectly separable. However, for 27 classes this is relatively good separation. If resume samples, represented by crosses, were semantically close to job descriptions, they could belong to the same job clusters. However, since the resume data has differences in the underlying distribution, some of its clusters are at least neighbours with corresponding job clusters, e.g., for Non-Profit, Computer/Internet, Arts, Retail and Engineering categories. We can not make any general conclusions about resume clusters due to the lack of data, but we can clearly find clusters for some categories, that are sometimes distant from corresponding job clusters. This suggests that the learned CNN representations are useful for resume classification as well, since clusters can be found using them.

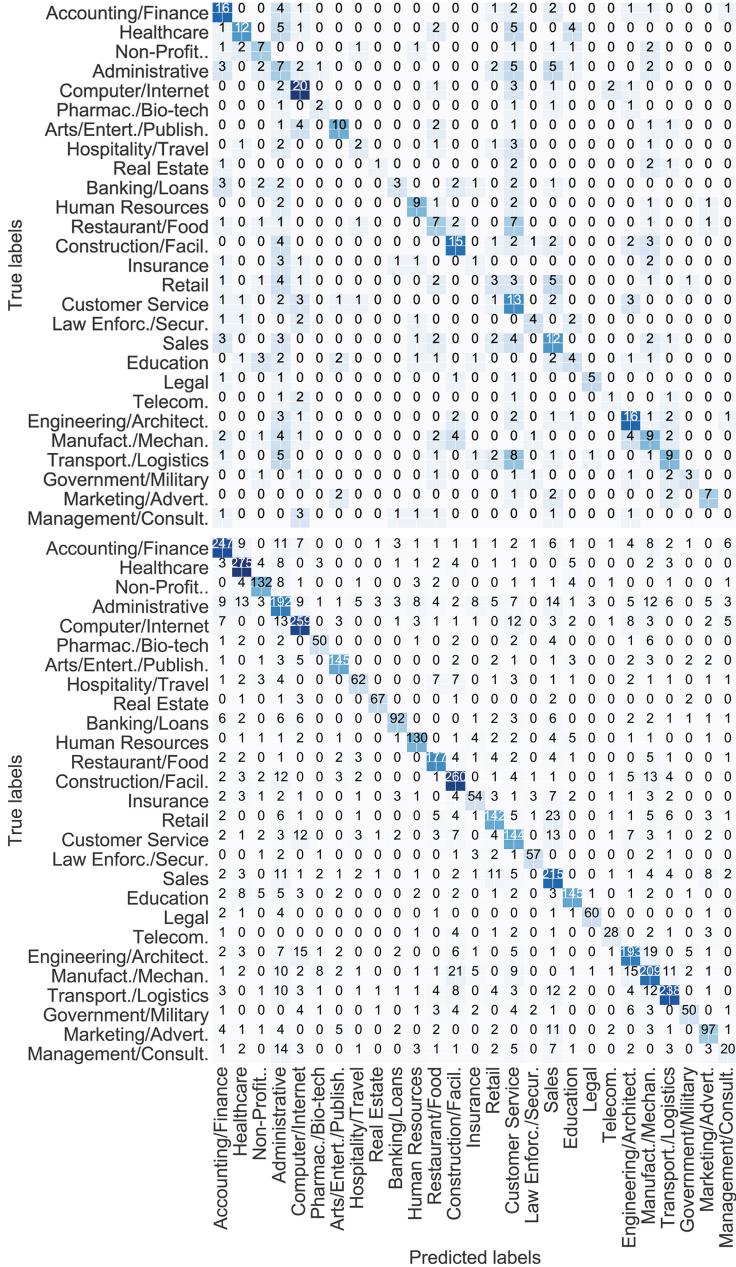


Fig. 3. Confusion matrix of resume (top) and job (bottom) classification results. In both datasets Management, Administrative, Customer Service, Retail and Manufacturing categories have a low recall. We assume that this happens due to the semantic closeness of these categories, since even a human can not always correctly make a clear distinction between them.

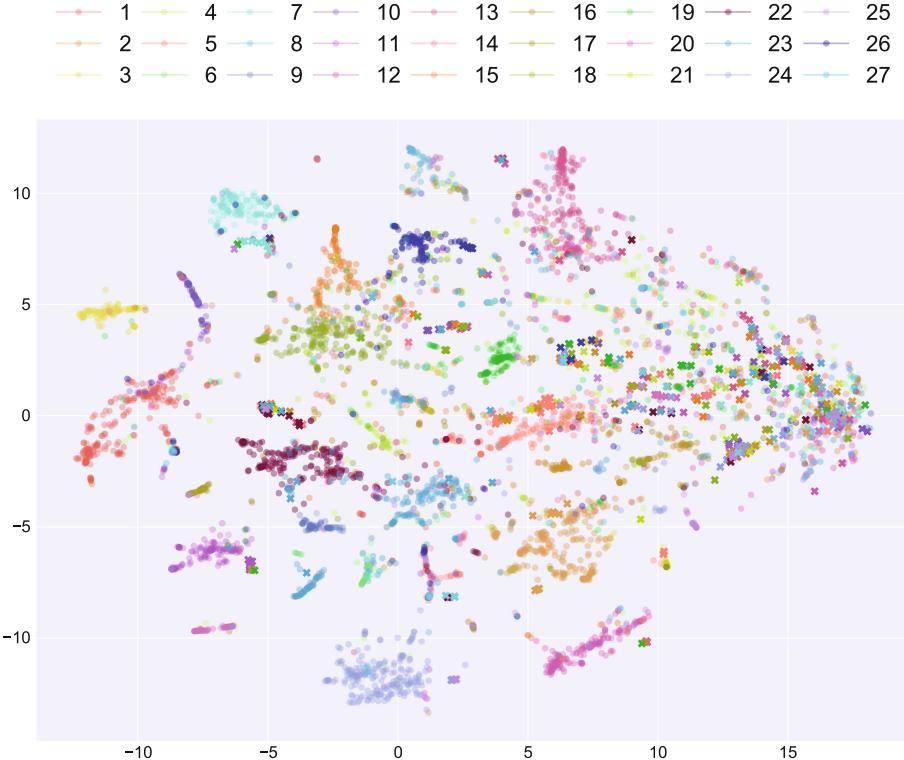


Fig. 4. t-SNE visualization using the CNN first layer outputs on job and resume data. We used all job (5,000) and resume (523) test data for fitting t-SNE. By visualizing vectors on 2D space, we check how useful are the representations learned by CNN to distinguish between the classes in familiar and new domains. There are 5000 test samples from job data, marked with circles, and 523 samples from resume data, marked with crosses, in total. We can observe the presence of category clusters formed by job samples, however they are not perfectly separable. Since the resume data has differences in underlying distribution, some resume clusters are neighbours with corresponding job clusters, e.g. from Non-Profit, Computer/Internet, Arts, Retail and Engineering categories. In fact, resume classes from these classes form neighbouring clusters or intersect with corresponding job clusters.

5 Conclusion

We have devised a resume classification method which is able to exploit the information contained in vast amounts labeled job description data in order to achieve higher accuracy. Since resumes are more sensitive data and difficult to obtain, compared to job summaries, we trained the proposed model only on job summaries and tested its performance on resume data with the same job category labels. A convolutional neural network for short text classification using word embeddings was trained and validated on 85,000 short job summaries mined from

Indeed. Then this network was used to classify a set of 523 candidate resumes and compared with a simple but effective **fastText** model. Our method achieved 74.88% accuracy on job classification task and 40.15 % on resume classification, thereby outperforming the existing **fastText** model by more than 6% on resume classification task and 3% on the job description task. Moreover, we applied our method to a small imbalanced dataset consisting of 98 children dream job descriptions. In this task CNN outperformed **fastText** by 22%.

Given the fact that no labels were used from resume data for training or validation, we consider CNN for short classification to be useful in a domain adaptation scenario. An interesting direction for future work would be to study whether the results can be improved by leveraging a small number of labeled resume samples to fine-tune the CNN model.

References

1. Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M.: Domain-adversarial neural networks. arXiv preprint [arXiv:1412.4446](https://arxiv.org/abs/1412.4446) (2014)
2. Al-Otaibi, S.T., Ykhlef, M.: A survey of job recommender systems. *Int. J. Phys. Sci.* **7**(29), 5127–5142 (2012)
3. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. *Mach. Learn.* **79**(1), 151–175 (2010)
4. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F., et al.: Analysis of representations for domain adaptation. In: *Advances in Neural Information Processing Systems*, vol. 19, p. 137 (2007)
5. Daume III, H., Marcu, D.: Domain adaptation for statistical classifiers. *J. Artif. Intell. Res.* **26**, 101–126 (2006)
6. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation, pp. 1180–1189 (2015)
7. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **17**(59), 1–35 (2016)
8. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: a deep learning approach. In: *International Conference on Machine Learning*, pp. 513–520 (2011)
9. Hong, W., Zheng, S., Wang, H., Shi, J.: A job recommender system based on user clustering. *J. Comput.* **8**(8), 1960–1967 (2013)
10. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint [arXiv:1607.01759](https://arxiv.org/abs/1607.01759) (2016)
11. Kim, Y.: Convolutional neural networks for sentence classification, pp. 1746–1751 (2014)
12. Kim, Y., Jernite, Y., Sontag, D., Rush, A.M.: Character-aware neural language models. In: *Thirtieth AAAI Conference on Artificial Intelligence*, pp. 2741–2749 (2016)
13. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: *International Conference on Machine Learning*, pp. 97–105 (2015)
14. Maaten, L.V.D., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008)

15. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
16. Tutubalina, E.: Dependency-based problem phrase extraction from user reviews of products. In: Král, P., Matoušek, V. (eds.) TSD 2015. LNCS (LNAI), vol. 9302, pp. 199–206. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24033-6_23
17. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in Neural Information Processing Systems, pp. 649–657 (2015)

Fighting with the Sparsity of Synonymy Dictionaries for Automatic Synset Induction

Dmitry Ustalov^{1,2}(✉), Mikhail Chernoskutov^{1,2}, Chris Biemann³,
and Alexander Panchenko³

¹ Ural Federal University, Yekaterinburg, Russia

{dmitry.ustalov,mikhail.chernoskutov}@urfu.ru

² Krasovskii Institute of Mathematics and Mechanics, Yekaterinburg, Russia

³ Universität Hamburg, Hamburg, Germany

{biemann,panchenko}@informatik.uni-hamburg.de

Abstract. Graph-based synset induction methods, such as MaxMax and WATSET, induce synsets by performing a global clustering of a synonymy graph. However, such methods are sensitive to the structure of the input synonymy graph: sparseness of the input dictionary can substantially reduce the quality of the extracted synsets. In this paper, we propose two different approaches designed to alleviate the incompleteness of the input dictionaries. The first one performs a pre-processing of the graph by adding missing edges, while the second one performs a post-processing by merging similar synset clusters. We evaluate these approaches on two datasets for the Russian language and discuss their impact on the performance of synset induction methods. Finally, we perform an extensive error analysis of each approach and discuss prominent alternative methods for coping with the problem of sparsity of the synonymy dictionaries.

Keywords: Lexical semantics · Word embeddings · Synset induction
Synonyms · Word sense induction · Synset induction
Sense embeddings

1 Introduction

A synonymy dictionary, representing synonymy relations between the individual words, can be modeled as an undirected graph where nodes are words and edges are synonymy relations.¹ Such a graph, called a synonymy graph or a synonymy network, tends to have a clustered structure [7]. This property is exploited by various graph-based word sense induction (WSI) methods, such as [23]. The goal of such WSI methods is to build a word sense inventory from various networks, such as synonymy graphs, co-occurrence graphs, graphs of distributionally related words, etc. (see a survey by Navigli [19]).

¹ In the context of this work, we assume that synonymy is a relation of lexical semantic equivalence which is context-independent, as opposed to “contextual synonyms” [32].

The clusters are densely connected subgraphs of synonymy graph that correspond to the groups of semantically equivalent words or synsets (sets of synonyms). Synsets are building blocks for WordNet [5] and similar lexical databases used in various applications, such as information retrieval [14]. Graph-based WSI combined with graph clustering makes it possible to induce synsets in an unsupervised way [12, 30]. However, these methods are highly sensitive to the structure of the input synonymy graph [30], which motivates the development of synonymy graph expansion methods.

In this paper, we are focused on the data sparseness reduction problem in the synonymy graphs. This problem is inherent to the majority of manually constructed lexical-semantic graphs due to the Zipf's law of word frequencies [33]: the long tail of rare words is inherently underrepresented. In this work, given a synonymy graph and a graph clustering algorithm, we compare the performance of two methods designed to improve synset induction. The goal of each method is to improve the final synset cluster structures. Both methods are based on the assumption that synonymy is a symmetric relation. We run our experiments on the Russian language using the WATSET state-of-the-art unsupervised synset induction method [30].

The contribution of this paper is a study of two principally different methods for dealing with the sparsity of the input synonymy graphs. The former, *relation transitivity* method, is based on expansion of the synonymy graph. The latter, *synset merging* method, is based on the mutual similarity of synsets.

2 Related Work

Hope and Keller [12] introduced the MaxMax clustering algorithm particularly designed for the word sense induction task. In a nutshell, pairs of nodes are grouped if they have a maximal mutual affinity. The algorithm starts by converting the undirected input graph into a directed graph by keeping the maximal affinity nodes of each node. Next, all nodes are marked as root nodes. Finally, for each root node, the following procedure is repeated: all transitive children of this root form a cluster and the root are marked as non-root nodes; a root node together with all its transitive children form a fuzzy cluster.

Van Dongen [3] presented the Markov Clustering (MCL) algorithm for graphs based on simulation of stochastic flow in graphs. MCL simulates random walks within a graph by alternation of two operators called expansion and inflation, which recompute the class labels. This approach has been successfully used for the word sense induction task [4].

Biemann [1] introduced Chinese Whispers, a clustering algorithm for weighted graphs that can be considered as a special case of MCL with a simplified class update step. At each iteration, the labels of all the nodes are updated according to the majority labels among the neighboring nodes. The author showed usefulness of the algorithm for induction of word senses based on corpus-induced graphs.

The ECO approach [8] was applied to induce a WordNet of the Portuguese language.² In its core, ECO is based on a clustering algorithm that was used to induce synsets from synonymy dictionaries. The algorithm starts by adding random noise to edge weights. Then, the approach applies Markov Clustering of this graph several times to estimate the probability of each word pair being in the same synset. Finally, candidate pairs over a certain threshold are added to output synsets.

In our experiments, we rely on the WATSET synset induction method [30] based on a graph meta-clustering algorithm that combines local and global hard clustering to obtain a fuzzy graph clustering. The authors shown that this approach outperforms all methods mentioned above on the synset induction task and therefore we use it as the strongest baseline to date.

Meyer and Gurevich [17] presented an approach for construction of an ontologized version of Wiktionary, by formation of ontological concepts and relationships between them from the ambiguous input dictionary, yet their approach does not involve graph clustering.

3 Two Approaches to Cope with Dictionary Sparseness

We propose two approaches for dealing with the incompleteness of the input synonymy dictionaries of a graph-based synset induction method, such as WATSET or MaxMax. First, we describe a graph-based approach that preprocesses the input graph by adding new edges. This step is applied *before* the synset induction clustering. Second, we describe an approach that post-processes the synsets by merging highly semantically related synsets. This step is applied *after* the synset induction clustering step, refining its results.

3.1 Expansion of Synonymy Graph via Relation Transitivity

Assuming that synonymy is an equivalence relation due to its reflexivity, symmetry, and transitivity, we can insert additional edges into the synonymy graph between nodes that are transitively, synonymous, i.e. are connected by a short path of synonymy links. We assume that if an edge for a pair of synonyms is missing, the graph still contains several relatively short paths connecting the nodes corresponding to these words.

Firstly, for each vertex, we extract its neighbors and the neighbors of these neighbors. Secondly, we compute the set of candidate edges by connecting the disconnected vertices. Then, we compute the number of simple paths between the vertices in candidate edges. Finally, we add an edge into the graph if there are at least k such paths which lengths are in the range $[i, j]$.

Particularly, the algorithm works as follows:

1. extract a first-order ego network N_1 and a second-order ego network N_2 for each node;

² <http://ontopt.dei.uc.pt>.

2. generate the set of candidate edges that connect the disconnected nodes in N_1 , i.e., the total number of the candidates is $C_{|N_1|}^2 - |E_{N_1}|$, where $C_{|N_1|}^2$ is the number of all 2-combinations over the $|N_1|$ -element set and E_{N_1} is the set of edges in N_1 ;
3. keep only those edge candidates that satisfy two conditions: (1) there are at least k paths p in N_2 , so no path contains the initial ego node, and (2) the length of each path belongs to the interval $[i, j]$.

The approach has two parameters: the minimal number of paths to consider k and the path length interval $[i, j]$. It should be noted that this approach processes the input synonymy graph without taking the polysemous words into account. Such words are then handled by the WATSET algorithm that induces word senses based on the expanded synonymy graph.

3.2 Synset Merging Based on Synset Vector Representations

We assume that closely related synsets carry equivalent meanings and use the following procedure to merge near-duplicate synsets:

1. learn synset embeddings for each synset using the SenseGram method by simply averaging word vectors that correspond to the words in the synset [26];
2. identify the closely related synsets using the $m\text{-}kNN$ algorithm [21] that considers two objects as closely related if they are mutual neighbors of each other;
3. merge the closely related synsets in a specific order: the smallest synsets are merged first, the largest are merged later; every synset can be merged only once in order to avoid giant merged clusters.

This approach has two parameters: the number of nearest neighbors to consider k (fixed to 10 in our experiments)³ and the maximal number of merged synsets t , e.g., if $t = 1$ then only the first mutual nearest neighbor is merged. It should be noted that this approach operates on synsets that which are already have been discovered by WATSET. Therefore, the merged synsets are composed of disambiguated word senses.

4 Evaluation

We evaluate the performance of the proposed approaches using the WATSET graph clustering method that shows state-of-the-art results on synset induction [30]. WATSET is a meta-clustering algorithm that disambiguates a (word) graph by first performing ego-network clustering to split nodes (words) into (word) senses. Then a global clustering is used to form (syn)sets of senses.

³ In general, the $m\text{-}kNN$ method can be parametrized by two different parameters: k_{ij} – the number of nearest neighbors from the word i to the word j and k_{ji} – the number of nearest neighbors from the word j to the word i . In our case, for simplicity, we set $k_{ij} = k_{ji} = k$.

For both clustering steps, any graph clustering algorithm can be employed; in [30], it was shown that combinations of Chinese Whispers [1] (CW) and Markov Clustering [3] (MCL) provide the best results. We also evaluated the same approaches with the MaxMax [12] method, but the results were virtually the same, so we omitted them for brevity.

4.1 Datasets

We evaluate the proposed augmentation approaches on two gold standard datasets for Russian: RuWordNet [15] and YARN [2]. Both are analogues of the original English WordNet [5].

We used the same input graph as in [30]; the graph is based on three synonymy dictionaries, the Russian Wiktionary, the Abramov’s dictionary and the UNLDC dictionary. The graph is weighted using the similarities from Russian Distributional Thesaurus (RDT) [24].⁴ To construct synset embeddings, we used word vectors from the RDT.

The lexicon of the input dictionary is different from the lexicon of RuWordNet [15], which includes a lot of domain-specific synsets. At the same time, the input dataset is the same as the data sources used for bootstrapping YARN [2].

The summary of the datasets is shown in Table 1: the “# words” column specifies the number of lexical units in the dataset (nodes of the input graph), the “# synonyms” column indicates the number of synonymy pairs appearing in the dataset (edges of the input graph). The problem of dictionary sparsity is the fact that some edges (synonyms) are missing in the input resource. Finally, the “# synsets” column specifies the number of resulting synsets (if applicable).

Table 1. Summary of the datasets used in the experiments.

Resource	# words	# synsets	# synonyms
Input Synonymy Dictionary: Wiktionary	83 092	n/a	211 986
Induced Synsets: WATSET MCL-MCL	83 092	36 217	406 430
Induced Synsets: WATSET CW-MCL	83 092	55 369	355 158
Gold Synsets: RuWordNet	110 242	49 492	278 381
Gold Synsets: YARN	9 141	2 210	48 291

4.2 Quality Measures

We report results according to standard word sense induction evaluation measures: paired precision, recall and F-score [16], i.e., each cluster of n words yields $\frac{n(n-1)}{2}$ synonymy pairs. The exact same evaluation protocol was used in the original WATSET publication. We perform evaluation on the intersection of gold standard lexicon and the lexicon of the induced resource.

⁴ <http://russe.nlpub.ru/downloads>.

4.3 Results

The evaluation results are shown in Fig. 1. As one may observe, in the case of the RuWordNet dataset, the method based on the transitivity expansion rendered almost no improvements in terms of recall while dramatically dropping the precision. The second method, based on synset embeddings shows much better results on this dataset: It substantially improves recall, yet at the cost of a drop in precision.

In case of the YARN dataset, the results are similar with the graph-based method significantly lagging behind the vector-based method. However, in this case, the difference in the observed performance is smaller with some configurations of the graph-based methods approaching the performance of the vector-based method. Similarly to the first dataset, both methods trade off gains in recall for the drops in precision. Note, however, that the vector-based method can perform a shift of the “sweet spot” of the clustering approach. While the F-measure remains at the same level, it is possible to obtain higher levels of recall, which can be useful for some applications. In the following section, we perform error analysis for each method.

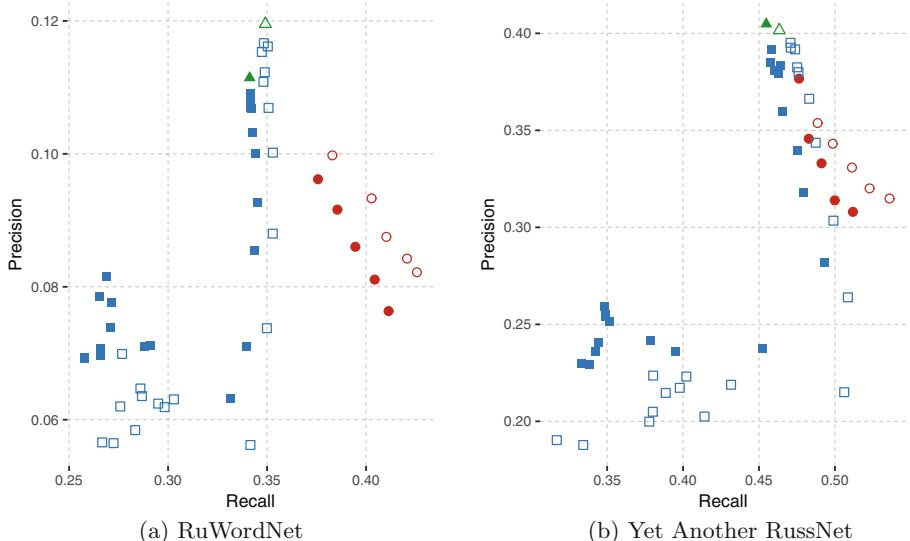


Fig. 1. Precision-recall plots built on two gold standard datasets for Russian. The shapes and colors: \blacktriangle original graph, \blacksquare transitivity expansion with the allowed path lengths $2 \leq i \leq j \leq 3$ and the number of simple paths $k \leq 10$, \bullet synset merging with the maximal number of merged mutual neighbours $t \in \{1, 2, 3, 5, 10\}$ and the number of nearest neighbours set to $k = 10$. Chinese Whispers-based WATSET configurations are hollow, while MCL-based are solid.

5 Discussion

Perfect synonyms are very rare, which is illustrated by the precision-recall plot in Fig. 1. Both methods insert relations of other types, such as association, co-hyponymy, hypernymy, etc. Recall increases with the level of inclusiveness of the configuration; this also causes significant drops in precision. The expansion methods presented in this paper could therefore be more useful for generation of other types of symmetric semantic relations, such as co-hyponymy.

5.1 Error Analysis: Synonymy Transitivity

We tried the following configurations of the approach: $2 \leq i \leq j \leq 3$, $k \leq 10$. However, only the variations with a small allowed length $i = j = 2$ and a high number of found simple paths $k \geq 5$ yielded viable results.

We explain the quick drops in precision by the fact that no word is a perfect synonym of another [10]. This results in the potential loss of the synonymy relation on each additional transitive node. While having a lot of pertinent edge insertions like “оказываться–появляться (show up – appear)” or “подтрунивать–стебаться (prank – make jokes)”, this method introduces such false positives like “кий–хлыст (cue – whip)”, “шеф–царь (boss – tsar)”, “солидный–корректный (solid – correct)”, etc.

One of the reasons of this outcome is that adding new edges increases the size of the communities. They capture neighboring vertices and edges belonging to other communities in the initial graph. Hence, on the one hand, we obtain communities with excess elements, while on the other hand, we observe depleted communities.

5.2 Error Analysis: Synset Merging

The different configurations of the vector-based method in this plot correspond to the following values of the t parameter (the maximum number of merged synsets): 1, 2, 3, 5, 10. Merging more than one synset at a time provides a substantial gain in the recall, yet again at the cost of the precision drop.

Table 2 presents an example of correct merging of synsets. The results of the clustering generate multiple small synsets that refer to the same meaning. Such synsets tend to be mutual nearest neighbors. Top ten most similar synsets to the synset “cynicism” are depicted. In this table, we also indicate whether each neighbor is the mutual nearest neighbor or not. In this example, the method of mutual nearest neighbors perfectly achieves its goal of merging synonymous synsets.

Table 3 presents an example of a wrong merging of synsets on the example of the synset “zinc, Zn”. This sample illustrates the reasons behind the drops in precision. While different chemical elements, such as zinc and cobalt are strongly semantically related, they are co-hyponyms of the common hypernym “chemical element”, and not synonyms, i.e. terms with equivalent meanings. This result is in line with the prior results showing that the majority of the nearest neighbors

Table 2. An example of correct synset merging, where predicted labels are equal to gold labels for the top $k = 10$ nearest neighbours of the synset “цинизм, циничность (cynicism, cynicism)”. In this table, the “Predicted” column contains mutually related synsets, while the “Gold” column lists expert judgments.

k	Similarity	Related Synset	Predicted	Gold
1	0.866	беспринципность, цинизм (unprincipledness, cynicism)	true	true
2	0.856	беспринципность, циничность (unprincipledness, cynicism)	true	true
3	0.853	кинизм, беспардонность, цинизм (cynicism, shamelessness, cynicism)	true	true
4	0.734	нахрапистость, нахальство, нахальность, циничность, бесцеремонность, нецеремонность (cheekiness, impudence, cheekiness, cynicism, brusqueness, unceremoniousness)	false	false
5	0.677	грубость, примитивизм (rudeness, primitivism)	false	false
6	0.677	хамство, лапидарность, хамёж, топорность, грубоность, прямолинейность (rudeness, conciseness, rudeness, clumsiness, rudeness, straightness)	false	false
7	0.674	безнравственность, беспринципность, злонравие, аморальность (wickedness, lack of principles, depravity, immorality)	false	false
8	0.671	бессстыдство, непристойность, бессовестность, нахрап (immorality, lack of principle, malice, immorality)	false	false
9	0.663	скепсис, скептичность (skepticism, skepticism)	true	false
10	0.661	фанатизм, ханжество (bigotry, bigotry)	false	false

Table 3. An example of wrong synset merging, where predicted labels are not equal to gold labels for the top $k = 10$ nearest neighbours of the synset “цинка, Zn (zinc, Zn)”. In this table, the “Predicted” column contains mutually related synsets, while the “Gold” column lists expert judgments

k	Similarity	Related Synset	Predicted	Gold
1	0.676	станинат кобальта, кобальт (cobalt stannate, cobalt)	true	false
2	0.673	Mg, магний (Mg, magnesium)	true	false
3	0.670	глиний, крылатый металл, алюминий, Al (clay, winged metal, aluminum, Al)	false	false
4	0.663	фосфор, P (phosphorus, P)	true	false
5	0.646	оксид, окись (oxide, oxide)	false	false
6	0.631	гидрооксид, гидроокись, гидроксид (hydroxide, hydroxide, hydroxide)	false	false
7	0.630	ванадий, V (vanadium, V)	true	false
8	0.628	рибофлавин, лактофлавин, витамин В (riboflavin, lactoflavin, vitamin B)	false	false
9	0.624	иодат, йодид, йодат (iodate, iodide, iodate)	false	false
10	0.618	кремний, Si (silicon, Si)	false	false

delivered by the distributional semantic models, such as the skip-gram model [18] used in our experiments, tend to be co-hyponyms as shown in prior studies [11, 20, 25, 31]. The results presented in both Tables 2 and 3 have been manually annotated by a single expert.

5.3 Other Ways to Deal with Sparseness of the Input Dictionary

In this section, we discuss avenues for future work: the prominent approaches that might be useful in addressing the sparseness of the synonym dictionaries.

Lexical-syntactic patterns for extraction of synonyms. Hearst patterns are widely used to mine hypernymy relations from text [27]. Such patterns can be also learned automatically [28, 29]. In [22], seven patterns for extraction of synonyms were proposed, which function in the same was as the Hearst patterns for hypernymy extraction. Such synonymy extraction patterns can be also learned automatically in the same fashion as patterns for hypernymy extraction from text [29]. Finally, hypernyms, antonyms, and other relations extracted from text can be used to filter our non-synonymous candidates.

Global clustering of synsets. It is possible to find groups of semantically related words (clique-like structures) and expand synonyms only within such communities with a graph clustering algorithm, such as Chinese Whispers [1]. The current graph-based transitivity expansion method does not consider the structure of the communities of the synonymy graph.

Synonymy detection as an anaphora resolution problem. Another observation is that synonyms are often not used in the same sentence, but instead used to ensure linguistic variance of the text. In this respect, synonymy extraction task is similar to the anaphora resolution task [13]. This line of work is related to prior work of [6] in detecting “bridging mentions”.

Crowdsourcing. Finally, the last option is simply to improve the quality of the input dictionaries by the means of crowdsourcing [9]. Namely, involving more people to edit Wiktionary that we use as the input data will increase the coverage of the extracted synsets, but large-scale crowdsourcing requires a set of elaborated quality control measures.

6 Conclusion

In this paper, we explored two alternative strategies for coping with the problem of inherent sparsity and incompleteness of the synonymy dictionaries. These sparsity issues hamper performance of the methods for automatic induction of synsets, such as MaxMax [12] and WATSET [30]. One of the proposed methods performs pre-processing of the graph of synonyms, while the second one performs post-processing of the induced synsets.

Our experiments on two large scale datasets show that (1) both methods are able to substantially improve recall, but at the cost of substantial drops of precision; (2) the post-processing approach yields better results overall. We conclude our study with an overview of prominent alternative approaches for expansion of incomplete synonymy dictionaries.

We believe the results of our study will be useful for both enriching the available lexical semantic resources like OntoWiktionary [17] as well as for increasing the lexical coverage of the input data for the graph-based word sense induction methods.

Acknowledgements. We acknowledge the support of the Deutsche Forschungsgemeinschaft (DFG) under the “JOIN-T” project, the DAAD, the RFBR under the projects no. 16-37-00203 мол_a and no. 16-37-00354 мол_a, and the RFH under the project no. 16-04-12019. The research was supported by the Ministry of Education and Science of the Russian Federation Agreement no. 02.A03.21.0006. The calculations were carried out using the supercomputer “Uran” at the Krasovskii Institute of Mathematics and Mechanics. Finally, we also thank four anonymous reviewers for their helpful comments.

References

1. Biemann, C.: Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In: Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing, pp. 73–80. TextGraphs-1. Association for Computational Linguistics, New York (2006)
2. Braslavski, P., Ustalov, D., Mukhin, M., Kiselev, Y.: YARN: spinning-in-progress. In: Proceedings of the 8th Global WordNet Conference (GWC 2016), pp. 58–65. Global WordNet Association, Bucharest (2016)
3. Van Dongen, S.: Graph Clustering by Flow Simulation. Ph.D. thesis, University of Utrecht (2000)
4. Dorow, B., Widdows, D.: Discovering corpus-specific word senses. In: Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics (EACL 2003), vol. 2, pp. 79–82. Association for Computational Linguistics, Budapest (2003)
5. Fellbaum, C.: WordNet: An Electronic Database. MIT Press, Cambridge (1998)
6. Feuerbach, T., Riedl, M., Biemann, C.: Distributional semantics for resolving bridging mentions. In: Proceedings of the International Conference Recent Advances in Natural Language Processing, pp. 192–199. INCOMA Ltd., Shoumen, Hissar (2015)
7. Gfeller, D., Chappelier, J.C., De Los Rios, P.: Synonym dictionary improvement through markov clustering and clustering stability. In: Proceedings of the International Symposium on Applied Stochastic Models and Data Analysis, pp. 106–113, Brest, France (2005)
8. Gonçalo Oliveira, H., Gomes, P.: ECO and Onto.PT: a flexible approach for creating a Portuguese wordnet automatically. Lang. Resour. Eval. **48**(2), 373–393 (2014)
9. Gurevych, I., Kim, J. (eds.): The People’s Web Meets NLP: Collaboratively Constructed Language Resources. Theory and Applications of Natural Language Processing. Springer, Heidelberg (2013). <https://doi.org/10.1007/978-3-642-35085-6>

10. Herrmann, D.J.: An old problem for the new psycho-semantics: synonymity. *Psychol. Bull.* **85**(3), 490–512 (1978)
11. Heylen, K., Peirsman, Y., Geeraerts, D., Speelman, D.: Modelling word similarity: an evaluation of automatic synonymy extraction algorithms. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008), pp. 3243–3249. European Language Resources Association, Marrakech (2008)
12. Hope, D., Keller, B.: MaxMax: a graph-based soft clustering algorithm applied to word sense induction. In: Gelbukh, A. (ed.) CICLing 2013. LNCS, vol. 7816, pp. 368–381. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37247-6_30
13. Lappin, S., Leass, H.J.: An algorithm for pronominal anaphora resolution. *Comput. Linguist.* **20**(4), 535–561 (1994)
14. Loukachevitch, N.V.: Thesauri in Information Retrieval Tasks. Moscow University Press, Moscow (2011). (in Russian)
15. Loukachevitch, N.V., Lashevich, G., Gerasimova, A.A., Ivanov, V.V., Dobrov, B.V.: Creating Russian wordnet by conversion. In: Computational Linguistics and Intellectual Technologies: Papers from the Annual Conference “Dialogue”, pp. 405–415. RSUH, Moscow (2016)
16. Manandhar, S., Klapaftis, I., Dligach, D., Pradhan, S.: SemEval-2010 Task 14: word sense induction & disambiguation. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 63–68. Association for Computational Linguistics, Uppsala (2010)
17. Meyer, C.M., Gurevych, I.: OntoWiktionary: Constructing an Ontology from the Collaborative Online Dictionary Wiktionary, pp. 131–161. IGI Global, Hershey (2012)
18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, vol. 26, pp. 3111–3119. Curran Associates Inc., Harrahs and Harveys (2013)
19. Navigli, R.: A quick tour of word sense disambiguation, induction and related approaches. In: Bieliková, M., Friedrich, G., Gottlob, G., Katzenbeisser, S., Turán, G. (eds.) SOFSEM 2012. LNCS, vol. 7147, pp. 115–129. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27660-6_10
20. Panchenko, A.: Comparison of the baseline knowledge-, corpus-, and web-based similarity measures for semantic relations extraction. In: Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics (GEMS 2011), pp. 11–21. Association for Computational Linguistics, Edinburgh (2011)
21. Panchenko, A., Adeykin, S., Romanov, A., Romanov, P.: Extraction of semantic relations between concepts with KNN algorithms on Wikipedia. In: Proceedings of the 2nd International Workshop on Concept Discovery in Unstructured Data, pp. 78–86, no. 871 in CEUR Workshop Proceedings, Leuven, Belgium (2012)
22. Panchenko, A., Morozova, O., Naets, H.: A semantic similarity measure based on lexico-syntactic patterns. In: Proceedings of KONVENS 2012, pp. 174–178, ÖGAI (2012)
23. Panchenko, A., Simon, J., Riedl, M., Biemann, C.: Noun sense induction and disambiguation using graph-based distributional semantics. In: Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016), pp. 192–202. Bochumer Linguistische Arbeitsberichte (2016)

24. Panchenko, A., Ustalov, D., Arefyev, N., Paperno, D., Konstantinova, N., Loukachevitch, N., Biemann, C.: Human and machine judgements for Russian semantic relatedness. In: Ignatov, D., et al. (eds.) AIST 2016. CCIS, vol. 661, pp. 221–235. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-52920-2_21
25. Peirsman, Y., Heylen, K., Speelman, D.: Putting things in order. First and second order context models for the calculation of semantic similarity. In: Proceedings of the 9th Journées internationales d’Analyse statistique des Données Textuelles (JADT 2008), pp. 907–916, Lyon, France (2008)
26. Pelevina, M., Arefyev, N., Biemann, C., Panchenko, A.: Making sense of word embeddings. In: Proceedings of the 1st Workshop on Representation Learning for NLP, pp. 174–183. Association for Computational Linguistics, Berlin (2016)
27. Seitner, J., Bizer, C., Eckert, K., Faralli, S., Meusel, R., Paulheim, H., Ponzetto, S.P.: A large database of hypernymy relations extracted from the web. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), pp. 360–367. European Language Resources Association (ELRA), Portorož (2016)
28. Shwartz, V., Goldberg, Y., Dagan, I.: Improving hypernymy detection with an integrated path-based and distributional method. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers, pp. 2389–2398. Association for Computational Linguistics, Berlin (2016)
29. Snow, R., Jurafsky, D., Ng, A.Y.: Learning syntactic patterns for automatic hypernym discovery. In: Proceedings of the 17th International Conference on Neural Information Processing Systems (NIPS 2004), pp. 1297–1304. MIT Press, Vancouver (2004)
30. Ustalov, D., Panchenko, A., Biemann, C.: Watset: automatic induction of synsets from a graph of synonyms. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers, pp. 1579–1590. Association for Computational Linguistics, Vancouver (2017)
31. Wandmacher, T.: How semantic is latent semantic analysis? In: Proceedings of RÉCITAL 2005. pp. 525–534, Dourdan, France (2005)
32. Zeng, X.M.: Semantic relationships between contextual synonyms. US-China Educ. Rev. 4(9), 33–37 (2007)
33. Zipf, G.K.: The Psycho-Biology of Language, Houghton, Mifflin, Oxford, England (1935)

Men Are from Mars, Women Are from Venus: Evaluation and Modelling of Verbal Associations

Ekaterina Vylomova¹(✉), Andrei Shcherbakov¹, Yury Philippovich²,
and Galina Cherkasova³

¹ The University of Melbourne, Melbourne, Australia
evylomova@gmail.com, ultrasparc@yandex.ru

² Moscow Polytech, Moscow, Russia
Y.philippovich@mail.ru

³ Institute of the Science of Language, Moscow, Russia
gacherk@mail.ru

Abstract. We present a quantitative analysis of human word association pairs and study the types of relations presented in the associations. We put our main focus on the correlation between response types and respondent characteristics such as occupation and gender by contrasting syntagmatic and paradigmatic associations. Finally, we propose a personalised distributed word association model and show the importance of incorporating demographic factors into the models commonly used in natural language processing.

Keywords: Associative experiments · Sociolinguistics
Language models · Word associations

1 Introduction

Most of contemporary approaches in natural language processing (NLP) mainly rely on well-annotated and clean textual corpora. For instance, language as well as translation models are typically trained over Europarl [12] or the Wall Street Journal corpora. As Eisenstein [5] noted, most of such corpora present language used by a very specific social group. For example, Hovy [7] showed that the models trained over the Wall Street Journal perform better for old language users. And it becomes extremely troublesome to adapt the models trained over these corpora to new domains such as Twitter. In most cases researchers either normalize the data (for instance, by using string and distributional similarity as in Han [6]) or apply various techniques of domain adaptation and knowledge transfer.

Recently several studies in sociolinguistics demonstrated how the NLP models could be improved by considering social factors (see Volkova [27], Stoop [25]). This inspired us to exploit associative experiments approach to demonstrate how specialization and gender might affect associations. In this paper we first propose

the dataset for associative pairs of Russian native speakers¹ and then show how association types vary across gender and occupation. We also present a simple PPMI-based model of associations and demonstrate the difference in the model’s predictions depending on the social characteristics.

The paper is structured as follows. We first discuss previously organized associative experiments, then we introduce the dataset for Russian speakers associations. In Sect. 4 we analyse how the associations depend on demographic factors and, finally, we present a personalised associative vector model.

2 Related Work

Introduced by Sir Francis Halton in 1870s, associative experiments became a common approach to study human cognition. Nowadays various researchers organized the experiments on many languages. Most of the experiments present English (American and British) native speakers (see Deese [4], Cramer [1], Kiss [11], Nelson [17]). De Groot [3] and De Deyne [2] conducted them for Dutch, and Shaps [22] for Swedish. There are also some for Eastern languages, such as Japanese (see Okamoto and Ishizaki [19] and Joyce [8]), Korean (Jung [9]); and Hebrew (Rubinstein [20]) for Semitic group. Lots of research had been done on Slavic languages as well. Novak [18] organized the experiment for Czech, Ufimtseva [26] presented Slavic Associative Thesaurus comprising of Russian, Belarusian, Bulgarian, and Ukrainian. Finally, Russian thesauri were developed by Leontiev [13] and Karaulov [10]. The latter one has been conducted in three stages during 1986–1997 and is one of the largest experiments. In addition to associations the dataset also contains demographic information such as age, gender, specialization, and location.

Most of the previous research had been focused on the study of reactions: their distribution and cross-lingual commonalities. Some of the researchers (e.g. Steyvers and Tenenbaum [24]) also studied the structure of human associative networks. They represented a network as a directed graph in which stimuli and reactions correspond to nodes whereas associations are edges connecting them. They showed that the node’s degree (the number of different reactions given for a stimulus) follows a power law distribution². In other words, there are several “hub” nodes with many connections and many “weak” nodes with small degree.

But very little had been done in terms of quantitative evaluation of demographic factors in associations. Current research fills up this gap. We investigate the reaction types distribution in regards to gender and speciality.

3 Dataset

The experiment conducted by Karaulov’s group, although being one of the most lasting ones, very quickly becomes outdated. Moreover, it has only been focused

¹ The dataset is available at <http://github.com/ivri/RusAssoc>.

² i.e. associative networks are scale-free.

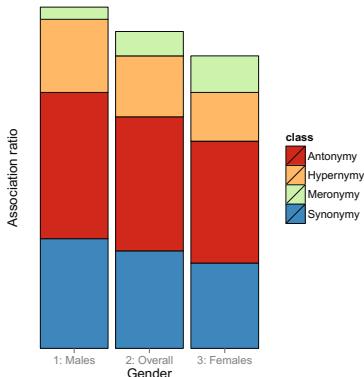


Fig. 1. Usage of various semantic relations across gender.

Table 1. An example list of associative pairs and corresponding n-grams along with relations.

Stimulus–reaction	Ngram freq.	Relations
yellow–colour	241	—
Russia–country	110	hyponymy
morning–good	445	—
mouth–face	6	part meronymy
medicine–clinic	0	domain
public–social	0	synonymy
help–find	33	—
most–outstanding	27	—
ask–answer	0	antonymy
here–there	18	antonymy
write–letter	218	—
impression–emotion	0	hyponymy

on the regions of Central Russia. To address these issues as well as to analyse the change of the associations over time, we additionally organized the associative experiments in various Russian regions, including Siberia and the Urals. The age of participants ranged from 16 to 26³, most of them were either undergraduate or postgraduate university students of ≈ 50 specialities. The experiments were organised as follows. A respondent received a questionnaire of 100 single-word stimuli. For each stimulus the respondent had to provide a reaction. There were no constraints on the reaction types, but the total time was limited to 10–15 minutes, i.e. the participants had 6–9 s for each stimulus. Most of the reactions appeared to be also single-word. Several association pairs are presented on Table 1.

In total, the dataset contains 4,997 questionnaires. The list of stimuli comprises of 1,213 various lemmas partially taken from Leontiev's list as well as most common reactions of previous Russian associative experiments from Karaulov [10]. The total number of different reactions received from the respondents is 50,359 (37,895 lemmas). Table 2 shows the top-10 most frequently used reactions. Surprisingly, we see a large overlap between current study and the experiment conducted in 1986–1997. Besides that, there is also an overlap with the most frequently used Russian words from Sharoff's list [23]. We also did not observe a significant cross-gender difference in the top reactions.

³ People in psycholinguistics typically assume that the core of the verbal associations becomes stable and does not significantly change after the age of 18.

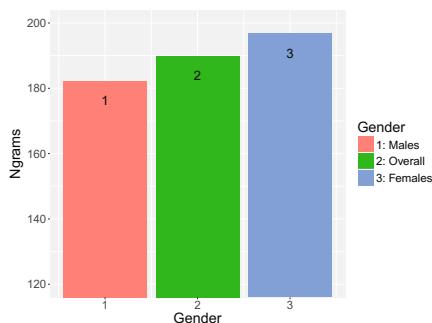


Fig. 2. Usage of ngrams by males and females.

Table 2. A top–10 list of the most frequent reactions received in the current study and Karaulov’s experiment compared to the top frequent words from Sharoff’s list.

Top–10	Karaulov’s	Sharoff’s
Human being	Human being	Year
Money	Home/House	Human being
Friend	Money	Time
Home/House	Day	Business
Life	Friend	Life
Day	Home	Day
World/Peace	Male	Hand
Big/Large	Fool	Work
Time	Business	Word
Child	Life	Place

4 Experiments

4.1 Association Types Analysis

Aiming to observe possible differences in distribution of association patterns among categories of respondents, we match associations against two major patterns as follows.

First, we check whether a stimulus – response pair matches an ngram observed in text corpora. We measure a smoothed sum of matched ngram frequencies:

$$S = \sum_{a \in A} \begin{cases} \log f(a), & \text{iff } f > 0, \\ 0, & \text{iff } f = 0 \end{cases} \quad (1)$$

where a is an association pair, A is a set of all association pairs, $f(a)$ is a corpus frequency of an ngram produced of a association.

We consider bigrams for single-word responses (the vast majority of cases). If we have two-word response, we match it against trigrams. Responses containing more than two words are treated as non-matching any known ngrams. We tried to match each association both in forward (*stimulus* → *response*) and backward (*response* → *stimulus*) direction, and each side (*stimulus* and *response*) was supplied both as is and in a lemmatized form.⁴ By doing that, we actually match each association against eight candidate ngrams, and we pick the maximum frequency observed over those ngrams. We used National Corpus of Russian Language⁵ as source for ngram frequencies.

Second, we extract associations that correspond to basic thesaurus relations such as synonyms, antonyms, hypernyms/hyponyms, meronyms/holonyms, and

⁴ We used *mystem* [21] to extract lemmas.

⁵ <http://www.ruscorpora.ru/corpora-freq.html>.

cause/effect. We use Russian WordNet⁶ [15] as our initial data source, where we count the number of matching associations for every relation type. Table 1 presents a list of associations and corresponding extracted ngram frequencies and relations.

We measure the values listed above as percentages to the total number of responses. We have done it for the full dataset and also for slices selected by respondent's gender or specialization.

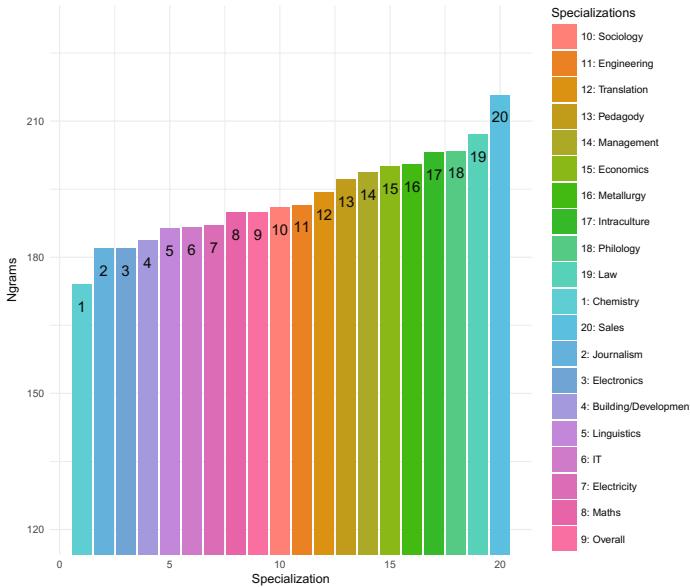


Fig. 3. Usage of Ngrams across various specializations.

Figures 2 and 1 show that men are more biased towards using semantically inspired associations (paradigmatic) whereas women are more likely to produce ngrams (syntagmatic)⁷. We observe a similar pattern (i.e. syntagmatic inversely related to paradigmatic) by looking at the specializations. Figure 3 presents S values for the top-20 most popular specializations. For instance, “chemistry” presents the highest scores in semantic relations whereas lower than average for ngrams. On the other hand, in the case of “sales” it is completely opposite. Note that most of the technical specializations and natural sciences demonstrate high scores for paradigmatic association types. We supposed that this is due to correlation between gender and occupation and the fact that gender still plays a significant role in the process of choosing the future career. In order to test that hypothesis, we calculated ngram usage figures normalized over gender (Fig. 4).

⁶ <http://wordnet.ru>.

⁷ With p-value < 0.001.

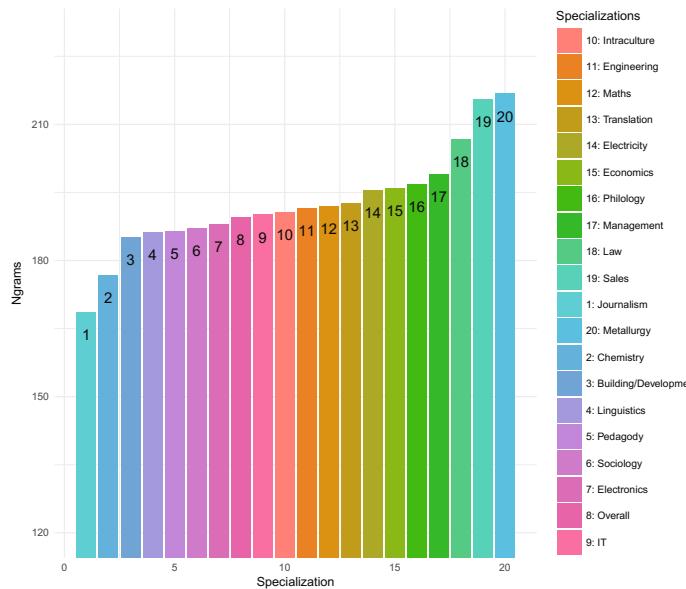


Fig. 4. Usage of Ngrams across various specializations. Normalised over the gender.

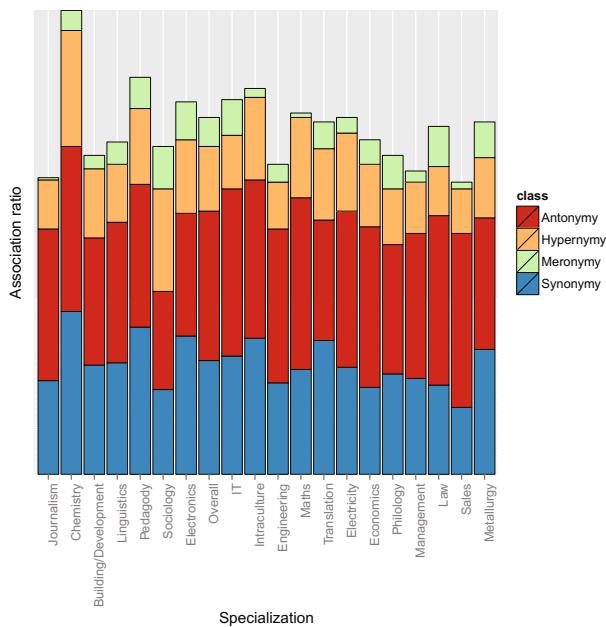


Fig. 5. Usage of various semantic relations across various specializations.

In this context, a normalized value is a half-sum of two corresponding average values, each one being computed over its respective respondent gender. The normalization didn't seem to smooth differences in ngram usage over various specializations. Therefore, one may conclude that the specialization standalone plays as a significant factor influencing word association patterns (Fig. 5).

4.2 Personalization of Vector Models

Now we turn to our experiments with associative vector models. We propose gender and specialization specific associative models. In order to create the models, we first slice the data by the proposed attribute. For instance, for “gender” we take two subsets corresponding to “male” and “female”, respectively.⁸ As we described earlier, their association frequency distributions present some differences which we would like our model to capture. Unlike traditional language modelling task, here we only rely on stimulus-reaction pair frequencies. Therefore, we consider SVD-PPMI [14] approach to get the distributed vector representations. The method had been shown to perform on par with neural models [14, 28], such as word2vec [16]. It is also less expensive in terms of the time complexity and better fits our task setting.⁹ We additionally train a baseline model on the full dataset to compare it to the personalised models. We would like to emphasize that usage of distributed vector models allows us to go beyond the scope of direct associations and generalize better.

Table 3 presents several examples for the top 10 nearest neighbours of “male”, “female” and baseline models. In this case, we observe mainly semantic differences. Notice a substantial variation in the predictions of the models if we provide them with “T” stimulus. Table 4 illustrates the models work for “sales” and “publishing” occupation types. In general, we find the ability to provide gender- and specialization-sensitive information useful for addressing the issues related to social language variation.

Table 5 additionally provides the difference in the model's predictions for two locations in Chelyabinsk oblast: a small town of Asha and an industrial city of Magnitogorsk.

Unfortunately, there is no consensus in the research community on how to evaluate the associative models. Most of the methods are based on direct comparison of statistical characteristics of the distributions of reactions each of the models generates. More over, to our knowledge, no theoretical framework or quality assessment or measures had been proposed for that so far. Therefore, we consider this part of research for our future studies.

⁸ The dataset is quite balanced and we have roughly the same number of questionnaires for both male and female participants.

⁹ We used the model implementation from <https://bitbucket.org/omerlevy/hyperwords>. We set the size of the context window to 1 (left and right words), embedding size to 100, context distribution smoothing of 0.75, token threshold value of 5, all the other parameters were left with their default values.

Table 3. A top–10 list of the nearest neighbours for each of the models.

Effectiveness			I			Work/Job		
All	Male	Female	All	Male	Female	All	Male	Female
result	usefulness	result	don't think	workaholic	ego	labour	labour	labour
practicality	result	process	stupid	bummer	you	well-paid	high-paid	effort
diligence	diligence	diligence	ego	lazy	individual	stock exchange	stock exchange	diligence
usefulness	practicality	quality	individual	quitter	he	deal	to work	ennoble
perspective	labour	science	nihilist	idler	we	ennoble	worker	deal
process	quality	perspective	Alex	pronoun	selfishness	succeed	deal	hard
quality	high paid	aspiration	Narcissus	student	everyone	effort	activity	worthy
stability	work/job	practicality	loser	sloven	myself	hard-working	office	workaholic
labour	utility	usefulness	nothingness	loafer	Narcissus	diligence	effort	salary
ambition	absolute	progress	selfish	sluggard	selfishness	to work	diligence	fervor

Table 4. A top–5 list of the nearest neighbours for “Sales” and “Publishing” specializations.

Time			Money			Red		
All	Sales	Publishing	All	Sales	Publishing	All	Sales	Publishing
second	together	last	spend	gold	income	October	color	anger
60	result	minute	deficit	a lot of fuel		square	sun	Lenin
minute	past	deficit	coin	wealth	import	orange	lamp	edge
hour	timely	class	cash	give	gas	Lenin	yellow	spite
clock	evening	long	tax	rich	penny	revolution	blue	revolution

Table 5. A top–5 list of the nearest neighbours for “Asha” and “Magnitogorsk” cities.

Time		Money		Red	
Asha	Magnitogorsk	Asha	Magnitogorsk	Asha	Magnitogorsk
of fame	boring	expensive	no money	colour	skin colour
second	waiting	to give	numbers	bright	green
clock	clock	salary	hope	white	yellow
no time	minute	income	debt	black	colour
back	train	to sell	change	blue	traffic lights

5 Conclusion

We presented a new dataset for Russian verbal associations. We also showed that social factors such as gender and specialization provide a significant amount of information on the type of association. Finally, we also proposed

a gender-sensitive associative model and demonstrated the significance of incorporating of social factors into the traditional NLP models.

Acknowledgments. We would like to thank all reviewers for their valuable comments and suggestions for future research directions. The first author was supported by the Melbourne International Research Scholarship (MIRS).

References

1. Cramer, P.: Word Association. Academic Press, New York (1968)
2. De Deyne, S., Storms, G.: Word associations: norms for 1,424 dutch words in a continuous task. *Behav. Res. Methods* **40**(1), 198–205 (2008)
3. de Groot, A.M.B.: Woordassociatiennormen met reactietijden. *Nederlands tijdschrift voor de psychologie en haar grensgebieden* **43**(6), 280–296 (1988)
4. Deese, J.: The Structure of Associations in Language and Thought. Johns Hopkins University Press, Baltimore (1966)
5. Eisenstein, J.: What to do about bad language on the internet. In: HLT-NAACL, pp. 359–369 (2013)
6. Han, B., Baldwin, T.: Lexical normalisation of short text messages: makn sens a# twitter. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 368–378. Association for Computational Linguistics (2011)
7. Hovy, D., Søgaard, A.: Tagging performance correlates with author age. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, pp. 483–488 (2015)
8. Joyce, T.: Constructing a large-scale database of Japanese word associations. *Glot-tometrics* **10**, 82–98 (2005). Corpus Studies on Japanese Kanji
9. Jung, J., Na, L., Akama, H.: Network analysis of Korean word associations. In: Proceedings of the NAACL HLT 2010 First Workshop on Computational Neuralinguistics, pp. 27–35. Association for Computational Linguistics (2010)
10. Karaulov, Y., Cherkasova, G., Ufimtseva, N., Sorokin, Y., Tarasov, E.: Russian associative thesaurus, Moscow (1998)
11. Kiss, G.R., Armstrong, C., Milroy, R., Piper, J.: An associative thesaurus of English and its computer analysis. In: The Computer and Literary Studies, pp. 153–165 (1973)
12. Koehn, P.: Europarl: a parallel corpus for statistical machine translation. In: MT Summit, vol. 5, pp. 79–86. Citeseer (2005)
13. Leontiev, A.: Norms of Russian word associations, Moscow (1977)
14. Levy, O., Goldberg, Y., Dagan, I.: Improving distributional similarity with lessons learned from word embeddings. *Trans. Assoc. Comput. Linguist.* **3**, 211–225 (2015)
15. Loukachevitch, N., Lashevich, G., Gerasimova, A., Ivanov, V., Dobrov, B.: Creating Russian wordnet by conversion. In: International Conference on Computational Linguistics and Intellectual Technologies Dialog 2016, pp. 405–415 (2016)
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of the Workshop at the International Conference on Learning Representations (2013)
17. Nelson, D.L., McEvoy, C.L., Schreiber, T.A.: The university of south orida free association, rhyme, and word fragment norms. *Behav. Res. Methods Instr. Comput.* **36**(3), 402–407 (2004)

18. Novák, Z.: Volné slovní parove asociace v cestine. Academia (1988)
19. Okamoto, J., Ishizaki, S.: Associative concept dictionary construction and its comparison with electronic concept dictionaries, pp. 214–220 (2001)
20. Rubinsten, O., Anaki, D., Henik, A., Drori, S., Faran, Y.: Free association norms in the Hebrew language. Word Norms in Hebrew, pp. 17–34 (2005)
21. Segalovich, I.: A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. In: MLMTA, pp. 273–280. Citeseer (2003)
22. Shaps, L.P., Johansson, B., Nilsson, L.: Swedish association norms (1976)
23. Sharoff, S.: The frequency dictionary for Russian (2001). Accessed 4 Dec 2007
24. Steyvers, M., Tenenbaum, J.B.: The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. *Cogn. Sci.* **29**(1), 41–78 (2005)
25. Stoop, W., van den Bosch, A.P.J.: Using idiolects and sociolects to improve word prediction. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pp. 318–327. Association for Computational Linguistics (2014)
26. Ufimmtseva, N., Cherkasova, G., Karaulov, Y., Evgenii, T.: Slavonic associative thesaurus: Russian, Belarussian, Bulgarian, Ukrainian. In: Problems of Applied Linguistics. 1 edn. (2004)
27. Volkova, S., Wilson, T., Yarowsky, D.: Exploring demographic language variations to improve multilingual sentiment analysis in social media. In: Proceedings of Empirical Methods on Natural Language Processing (EMNLP 2013), pp. 1815–1827 (2013)
28. Vylomova, E., Rimmel, L., Cohn, T., Baldwin, T.: Take and took, gaggle and goose, book and read: evaluating the utility of vector differences for lexical relation learning. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016), pp. 1671–1682 (2016)

Rotations and Interpretability of Word Embeddings: The Case of the Russian Language

Alexey Zobnin^(✉)

National Research University Higher School of Economics, Moscow, Russia
azobnin@hse.ru

Abstract. Consider a continuous word embedding model. Usually, the cosines between word vectors are used as a measure of similarity of words. These cosines do not change under orthogonal transformations of the embedding space. We demonstrate that, using some canonical orthogonal transformations from SVD, it is possible both to increase the meaning of some components and to make the components more stable under re-learning. We study the interpretability of components for publicly available models for the Russian language (RusVectōrēs, fastText, RDT).

1 Introduction

Word embeddings are frequently used in NLP tasks. In vector space models every word from the source corpus is represented by a dense vector in \mathbb{R}^d , where the typical dimension d varies from tens to hundreds. Such embedding maps similar (in some sense) words to close vectors. These models are based on the so called distributional hypothesis: similar words tend to occur in similar contexts [11]. Some models also use letter trigrams or additional word properties such as morphological tags.

There are two basic approaches to the construction of word embeddings. The first is count-based, or explicit [8, 20]. For every word-context pair some measure of their proximity (such as frequency or PMI) is calculated. Thus, every word obtains a sparse vector of high dimension. Further, the dimension is reduced using singular value decomposition (SVD) or non-negative sparse embedding (NNSE). It was shown that truncated SVD or NNSE captures latent meaning in such models [17, 25]. That is why the components of embeddings in such models are already in some sense canonical. The second approach is prediction-based, or implicit. Here the embeddings are constructed by a neural network. Popular models of this kind include word2vec [22, 23] and fastText [5].

Consider a prediction-based word embedding model. Usually in such models two kinds of vectors, both for words and contexts, are constructed. Let N be the vocabulary size and d be the dimension of embeddings. Let W and C be $N \times d$ -matrices whose rows are word and context vectors. As a rule, the objectives of such models depend on the dot products of word and context vectors, i. e., on the elements of WC^T . In some models the optimization can be directly rewritten as a matrix factorization problem [7, 19]. This matrix remains unchanged under

substitutions $W \mapsto WS$, $C \mapsto CS^{-1}^T$ for any invertible S . Thus, when no other constraints are specified, there are infinitely many equivalent solutions [9].

Choosing a good, not necessarily orthogonal, post-processing transformation S that improves quality in applied problems is itself interesting enough [24]. However, only word vectors are typically used in practice, and context vectors are ignored. The cosine distance between word vectors is used as a similarity measure between words. These cosines will not change if and only if the transformation S is orthogonal. Such transformations do not affect the quality of the model, but may elucidate the meaning of vectors' components. Thus, the following problem arises: *what orthogonal transformation is the best one for describing the meaning of some (or all) components?*

It is believed that the meaning of the components of word vectors is hidden [10]. But even if we determine the “meaning” of some component, we may lose it after re-training because of random initialization, thread synchronization issues, etc. Many researchers [2, 12, 21, 32] ignore this fact and, say, work with vector components directly, and only some of them take basis rotations into account [34]. We show that, generally, re-trained model differ from the source model by almost orthogonal transformation. This leads us to the following problem: *how one can choose the canonical coordinates for embeddings that are (almost) invariant with respect to re-training?*

We suggest using well-known plain old technique, namely, the singular value decomposition of the word matrix W . We study the principal components of different models for Russian language (RusVectōrēs, RDT, fastText, etc.), although the results are applicable for any language as well.

2 Related Work

Interpretability of the components have been extensively studied for topic models. In [6, 18] two methods for estimating the coherence of topic models with manual tagging have been proposed: namely, word intrusion and topic intrusion. Automatic measures of coherence based on different similarities of words were proposed in [1, 27]. But unlike topic models, these methods cannot be applied directly to word vectors.

There are lots of new models where interpretability is either taken into account by design [21] (modified skip-gram that produces non-negative entries), or is obtained automagically [2] (sparse autoencoding).

Lots of authors try to extract some predefined significant properties from vectors: [12] (for non-negative sparse embeddings), [34] (using a CCA-based alignment between word vectors and manually-annotated linguistic resource), [31] (ultradense projections).

Singular vector decomposition is the core of count-based models. To our knowledge, the only paper where SVD was applied to prediction-based word embedding matrices is [24]. In [4] the first principal component is constructed for sentence embedding matrix (this component is excluded as the common one).

Word embeddings for Russian language were studied in [3, 14, 15, 28].

3 Theoretical Considerations

3.1 Singular Value Decomposition

Let $m \geq n$. Recall [13] that a singular value decomposition (SVD) of an $m \times n$ -matrix M is a decomposition $M = U\Sigma V^T$, where U is an $m \times n$ matrix, $U^T U = I_n$, Σ is a diagonal $n \times n$ -matrix, and V is an $n \times n$ orthogonal matrix. Diagonal elements of Σ are non-negative and are called singular values. Columns of U are eigenvectors of MM^T , and columns of V are eigenvectors of $M^T M$. Squares of singular values are eigenvalues of these matrices. If all singular values are different and positive, then SVD is unique up to permutation of singular values and choosing the direction of singular vectors. But if some singular values coincide or equal zero, new degrees of freedom arise.

3.2 Invariance Under Re-training

Learning methods are usually not deterministic. The model re-trained with similar hyperparameters may have completely different components. Let M_1 and M_2 be the word matrices obtained after two separate trainings of the model. Let these embeddings be similar in the sense that cosine distances between words are almost the same, i. e., $M_1 M_1^T \approx M_2 M_2^T$. Suppose also that singular values of each M_i are different and non-zero. Then one can show that M_1 and M_2 differ only by the (almost) orthogonal factor. Indeed, left singular vectors in SVD of M_i are eigenvectors of $M_i M_i^T$. Hence, matrices U and Σ in SVD of M_1 and M_2 can be chosen the same. Thus, $M_2 \approx M_1 Q$, where $Q Q^T = I_d$. Here Q can be chosen as $V_1 V_2^T$ where V_i are matrices of right singular vectors in SVD of M_i .

3.3 Interpretability Measures

One of traditional measures of interpretability in topic modeling looks as follows [18, 26]. For each component, n most probable words are selected. Then for each pair of selected words some co-occurrence measure such as PMI is calculated. These values are averaged over all pairs of selected words and all components. The other approaches use human markup. Such measures need additional data, and it is difficult to study them algebraically. Also, unlike topic modeling, word embeddings are not probabilistic: both positive and negative values of coordinates should be considered.

Let all word vectors be normalized and W be the word matrix. Inspired by [27], where vector space models are used for evaluating topic coherence, we suggest to estimate the interpretability of k th component as

$$\text{interp}_k W = \sum_{i,j=1}^N W_{i,k} W_{j,k} (W_i \cdot W_j).$$

The factors $W_{i,k}$ and $W_{j,k}$ are the values of k th components of i th and j th words. The dot product $(W_i \cdot W_j)$ reflects the similarity of words. Thus, this measure will be high if similar words have similar values of k th coordinates.

What orthogonal transformation Q maximizes this interpretability (for some, or all components) of WQ ? In matrix terms,

$$\text{interp}_k W = (W^T W W^T W)_{k,k},$$

and

$$\text{interp}_k WQ = (Q^T W^T W W^T W Q)_{k,k}$$

because Q is orthogonal. The total interpretability over all components is

$$\begin{aligned} \sum_{k=1}^d \text{interp}_k WQ &= \sum_{k=1}^d (Q^T W^T W W^T W Q)_{k,k} \\ &= \text{tr } Q^T W^T W W^T W Q = \text{tr } (W^T W W^T W) = \sum_{k=1}^d \text{interp}_k W, \end{aligned}$$

because $\text{tr } Q^T X Q = \text{tr } Q^{-1} X Q = \text{tr } X$. It turns out that *in average* the interpretability is constant under any orthogonal transformation. But it is possible to make the first components more interpretable due to the other components. For example,

$$(Q^T W^T W W^T W Q)_{1,1} = (q^T W^T W q)^2$$

is maximized when q is the eigenvector of $W^T W$ with the largest singular value, i. e., the first right singular vector of W [13]. Let's fix this vector and choose other vectors to be orthogonal to the selected ones and to maximize the interpretability. We arrive at $Q = V$, where V is the right orthogonal factor in SVD $W = U \Sigma V^T$.

4 Experiments

4.1 Canonical Basis for Embeddings

We train two fastText skipgram models on the Russian Wikipedia with default parameters. First, we normalize all word vectors. Then we build SVD decompositions¹ of obtained word matrices and use V as an orthogonal transformation. Thus, new “rotated” word vectors are described by the matrix $WV = U\Sigma$. The corresponding singular values are shown in Fig. 1, they almost coincide for both models (and thus are shown only for the one model). For each component both in the source and the rotated models we take top 50 words with maximal (positive) and bottom 50 words with minimal (negative) values of the component. Taking into account that principal components are determined up to the direction, we join these positive and negative sets together for each component.

¹ With `numpy.linalg.svd` it took up to several minutes for 100 K vocabulary.

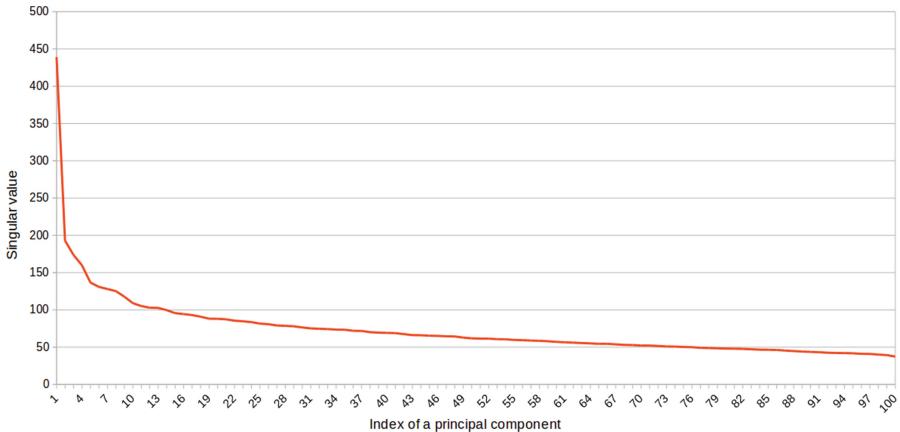


Fig. 1. Decreasing of singular values for the rotated fastText models (dim = 100).

We measure the overlapping of these sets of words. Additionally, we use the following alignment of components: first, we look for the free indices i and j such that i th set of words from the first model and j th set of words from the second model have the maximal intersection, and so on. We call the difference $i - j$ the alignment shift for the i th component. Results are presented in Figs. 2 and 3.

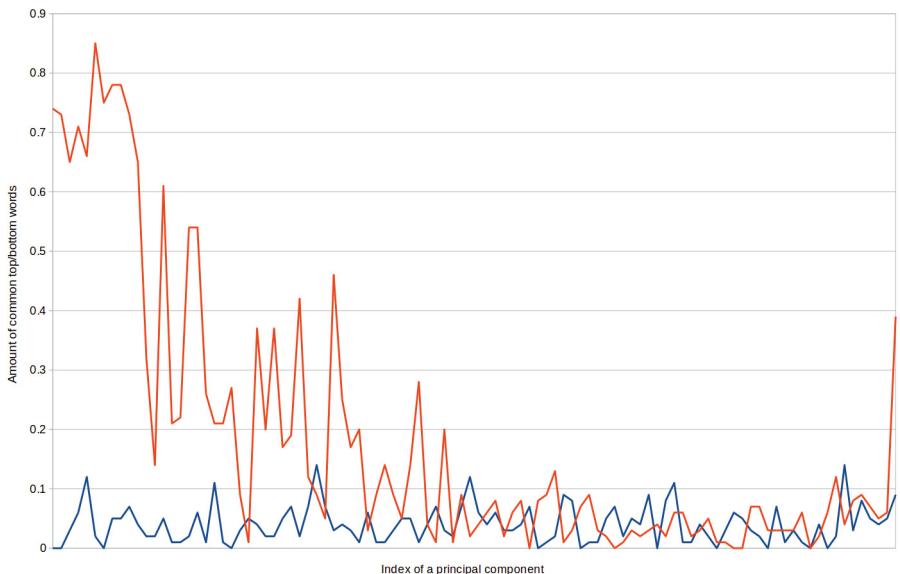


Fig. 2. The amount of common top and bottom words for the source models (blue) and the rotated models (red). (Color figure online)

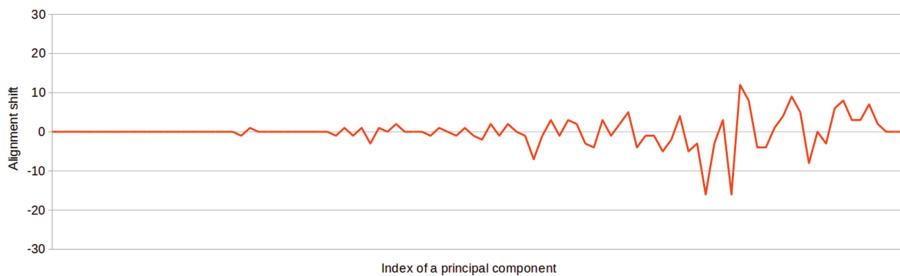


Fig. 3. Alignment shifts for the rotated models.

We see that at least for the first part of principal components (in the rotated models) the overlapping is big enough and is much larger than for the source models. Moreover, these first components have almost zero alignment shifts. Other principal components have very similar singular values, and thus they cannot be determined uniquely with high confidence.

Normalized interpretability measures for different components (calculated for 50 top/bottom words) for the source and the rotated models are shown in Fig. 4.

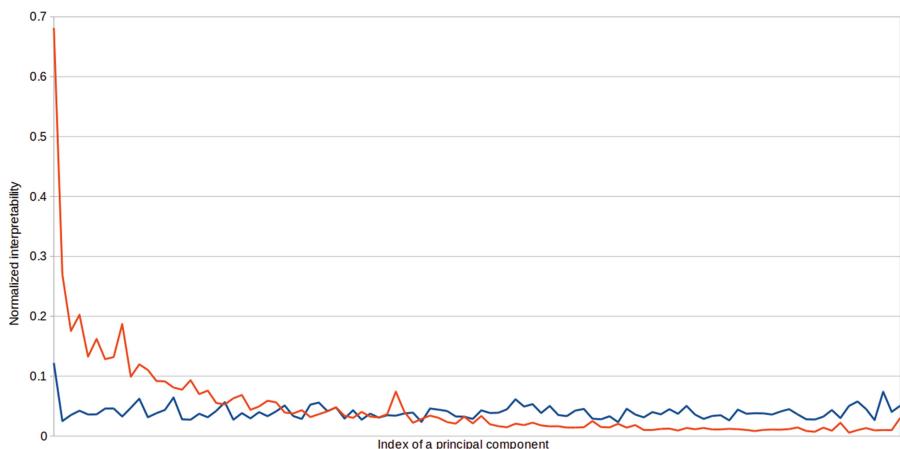


Fig. 4. Normalized interpretability values for different components calculated on top/bottom 50 words for each component in source coordinates (blue) and principal coordinates (red). (Color figure online)

4.2 Principal Components of Different Models

We took the following already published models:

- RusVectōrēs² lemmatized models (actually, word2vec) trained on different Russian corpora [16];
- Russian Distributional Thesaurus³ (actually, word2vec skipgram) models trained on Russian books corpus [29];
- fastText⁴ model trained on Russian Wikipedia [5].

For each model we took $n = 10000$ or $n = 100000$ most frequent words. Each word vector was normalized in order to replace cosines with dot products. Then we perform SVD $W = U\Sigma V^T$ and take the matrix $WV = U\Sigma$. For each of d components we sort the words by its value and choose top t “positive” and bottom t “negative” words ($t = 15$ or 30). For clarity, every selection was clustered into buckets with the simplest greedy algorithm: list the selected words in decreasing order of frequency and either add the current word to some cluster if it is close enough to the word (say, the cosine is greater than 0.6), or make a new cluster. The cluster’s vector is the average vector of its words. Intuitively, the smaller the number of clusters, the more interpretable the component is. Similar approach was used in [30].

Tables 1, 2 and 3 in the Appendix show the top “negative” and “positive” words of the first principal components for different models. We underline that principal components are determined up to the direction, and thus the separation into “negative” and “positive” parts is random. The full results are available at <https://alzobnin.github.io/>. We cluster these words as described above; different clusters are separated by semicolons. We see the following interesting features in the components:

- stop words: prepositions, conjunctions, etc. (RDT 1, fastText 1; in RusVectōrēs models they are absent just because they were filtered out before training);
- foreign words with separation into languages (fastText 2, web 2), words with special orthography or tokens in broken encoding (not presented here);
- names and surnames (RDT 8, fastText 3, web 3), including foreing names (fastText 9, web 6);
- toponyms (not presented here) and toponym descriptors (web 7);
- fairy tale characters (fastText 6);
- parts of speech and morphological forms (cases and numbers of nouns and adjectives, tenses of verbs);

² <http://rusvectores.org/ru/models/>.

³ https://nlp.ru/Russian_Distributional_Thesaurus.

⁴ <https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md>.

- capitalization (in fact, first positions in the sentences) and punctuation issues (e. g., non-breaking spaces);
- Wikipedia authors and words from Wikipedia discussion pages (fastText 5);
- other different semantic categories.

We also made an attempt to describe obtained components automatically in terms of common contexts of common morphological and semantic tags using MyStem tagger and semantic markup from Russian National Corpus. Unfortunately, these descriptions are not as good as desired and thus they are not presented here.

5 Conclusion

We study principal components of publicly available word embedding models for the Russian language. We see that the first principal components indeed are good interpretable. Also, we show that these components are almost invariant under re-learning. It will be interesting to explore the regularities in canonical components between different models (such as CBOW versus Skip-Gram, different train corpora and different languages [33]. It is also worth to compare our intrinsic interpretability measure with human judgements.

Acknowledgements. The author is grateful to Mikhail Dektyarev, Mikhail Nokel, Anna Potapenko and Daniil Tararukhin for valuable and fruitful discussions.

Appendix

Top/bottom words for the first few principal components for different Russian models

Table 1. RDT model, dim = 100, 10 K most frequent words

1	, не что как но то так же еще только уже даже того теперь действительно
2	деятельности отношении различных следовательно основе частности е отдельных основных посредством рамках данного определенных значительной возникновения обернулся прошептал оглянулся тихонько испуганно позвал присел крикнула повернувшись хрюпло вскрикнула обернувшись оглянулась позвала на-гнулся
3	тебе могу хочу понимаю скажу скажи правду сомневаюсь считаешь считаете подумай поверь согласна согласится стены вдоль справа слева видны колонны полосы виднелись посередине вы-сотой рядами бокам
4	приказал приказ срочно прибыл штаб отправил потребовал доложил напра-вил распоряжение распорядился выехал любовь любви душа страсти природа страсть красоты красота красоту че-ловеческая печаль; человеческой плоти человеческое
5	вечер вечера кафе обеда позвонила ресторане отеле вечерам проводила утрам; приехала отправилась ходила мамой купила меч воин меча клинок копье взмахнул; рявкнул вскинул выкрикнул проши-пел дернулся вскрикнул завопил прорычал
6	предмет предмета; компьютер автоматически компьютера; рассматривать модели модель анализа включает; клиента клиент войны враги войско волки лесах; деревню родину родные; боялись старики; умирать погибнуть
7	получается короче небось блин нету хрен кой; работают умеют берут платят; штук поменьше гнев отчаяние волнение отчаяния испытывала охватило; покинул встретил-ся застал покинула; объятиях страстью
8	николай пётр павел иванович михаил василий григорий васильевич михай-лович георгий федорович сможет смогу смогла готова шанс попыталась способна пытаться; выбраться вырваться выжить сопротивляться убежать сбежать
9	опять иван ваня алеша; начинается открывается следующая; москва петер-бург киев; весна осень мужчины мужчин; воины эльфы; казались выглядели представляли напо-минали являлись отличались позволяли держались
10	хлеб хлеба; посуду ложку видел встречались; произошло творится находясь; находится существует знаем; планета станция

Table 2. fastText model, 100 K most frequent words

Table 3. RusVectōrēs web model, 100K most frequent words

	информация _{noun} услуга _{noun} предложение _{noun} оплата _{noun} получение _{noun} законодательство _{noun} размещение _{noun} работодатель _{noun} заинтересованный _{adj} трудоустройство _{noun} соискатель _{noun} ; условие _{noun} необходимый _{adj} независимо _{adv}
1	анонсирование::плэйкастовый _{noun} непросмотренный::резюме _{noun} tools _{:trade} _{noun} webkind _{noun} support _{:info} _{noun} yellcity _{noun} elec _{::elec} _{noun} spell _{::correction} _{noun} МИКОЛЬ::ГОГОЛЬ _{noun} ненормовані _{noun} электроника _{::tech} _{home} _{noun} copyright _{::restate} _{noun} fannet _{::org} _{noun} своб _{::индексир} _{noun} ted _{::lapidus} _{noun}
2	value _{noun} plus _{noun} classic _{noun} super _{noun} light _{noun} series _{noun} techn _{noun} standard _{noun} horizon _{noun} cyber _{noun} regular _{noun} circuit _{noun} isis _{noun} ; blue _{noun} gold _{noun} сказать _{verb} знать _{verb} говорить _{verb} приходить _{verb} спрашивать _{verb} пойти _{verb} по-думать _{verb} решаться _{verb} удивляться _{verb} припомнить _{verb} впрямь _{adv} недоумевать _{verb} сговариваться _{verb} отчего-то _{adv} помалкивать _{verb}
3	сделать _{verb} делать _{verb} забывать _{verb} просто _{adv} угодно _{part} надоедать _{verb} любобой _{pron} пугаться _{verb} чертовски _{adv} may _{::captain} _{noun} черт _{::побрать} _{verb} ; смотреть _{verb} посмотреть _{verb} попов _{noun} андреев _{noun} калинин _{noun} максимов _{noun} мельников _{noun} тихомиров _{noun} емельянов _{noun} кондратьев _{noun} румянцев _{noun} романовский _{noun} андрианов _{noun} чеботарев _{noun} горячев _{noun} моисеенко _{noun} чудновский _{noun}
4	огнетушитель _{noun} балоны _{noun} баллон _{noun} закрутка _{noun} грузик _{noun} объем _{noun} приспособы _{noun} воздушка _{noun} акуратно _{adv} собраный _{adj} ; железка _{noun} ессыный _{adj} британский _{adj} роберт _{noun} джордж _{noun} известность _{noun} влиятельный _{adj} габриэль _{noun} фрэнсис _{noun} коэн _{noun} чарлз _{noun} гарольд _{noun} эдмунд _{noun} фредерика _{noun} тэтчер _{noun} теодора _{noun} джулиана _{noun}
5	образ _{noun} лицъ _{part} род _{noun} человеческий _{adj} глубокий _{adj} прежде _{adv} естественный _{adj} характерный _{adj} подобно _{adv} по-видимому _{adv} отчетливый _{adj} рыбы _{noun} рый _{noun} редуцированный _{adj} позвонить _{verb} звонить _{verb} ; присылат _{verb} отсылать _{verb} ; привет _{noun} личка _{noun} зарегистрировать _{verb} зарегистрированный _{adj}
6	чаяние _{noun} суетный _{adj} обличие _{noun} безбожный _{adj} своеволие _{noun} леность _{noun} властолюбие _{noun} чуждаться _{verb} иоаннов _{noun} вековечный _{adj} юродство _{noun} пит _{noun} бред _{noun} бишоп _{noun} пирсон _{noun} куин _{noun} филд _{noun} моск _{noun} кроуфорд _{noun} дафф _{noun} уолтерс _{noun} дэйли _{noun} слоун _{noun} роуч _{noun} макинтайр _{noun}
7	город _{noun} дом _{noun} улица _{noun} парк _{noun} столица _{noun} дворец _{noun} городок _{noun} недалеко _{adv} холм _{noun} неподалеку _{adv} пригород _{noun} близ _{adv} окрестности _{noun} подножие _{noun} адекватный _{adj} адекватно _{adv} неадекватный _{adj} адекватность _{noun} априори _{adv} латентный _{adj} неадекватность _{noun} нормальность _{noun} лакмусовый::бумажка _{noun} когнитивный::диссонанс _{noun}

Note misspellings in 4a.

References

1. Aletras, N., Stevenson, M.: Evaluating topic coherence using distributional semantics. In: Proceedings of IWCS 2013, pp. 13–22 (2013)
2. Andrews, M.: Compressing word embeddings. In: Hirose, A., Ozawa, S., Doya, K., Ikeda, K., Lee, M., Liu, D. (eds.) ICONIP 2016. LNCS, vol. 9950, pp. 413–422. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46681-1_50
3. Arefyev, N., Panchenko, A., Lukin, A., Lesota, O., Romanov, P.: Evaluating three corpus-based semantic similarity systems for Russian. In: Dialogue (2015)
4. Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings. In: ICLR (2017)
5. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. [arXiv:1607.04606](https://arxiv.org/abs/1607.04606) (2016)
6. Chang, J., Boyd-Graber, J.L., Gerrish, S., Wang, C., Blei, D.M.: Reading tea leaves: how humans interpret topic models. In: Nips, vol. 31, pp. 1–9 (2009)
7. Cotterell, R., Poliak, A., Van Durme, B., Eisner, J.: Explaining and generalizing skip-gram through exponential family principal component analysis. In: EACL 2017, p. 175 (2017)
8. Dhillon, P.S., Foster, D.P., Ungar, L.H.: Eigenwords: spectral word embeddings. *J. Mach. Learn. Res.* **16**, 3035–3078 (2015)
9. Fonarev, A., Hrinchuk, O., Gusev, G., Serdyukov, P., Oseledets, I.: Riemannian optimization for skip-gram negative sampling. [arXiv:1704.08059](https://arxiv.org/abs/1704.08059) (2017)
10. Gladkova, A., Drozd, A., Center, C.: Intrinsic evaluations of word embeddings: what can we do better? In: 1st Workshop on Evaluating Vector Space Representations for NLP, pp. 36–42 (2016)
11. Harris, Z.S.: Distributional structure. *Word* **10**(2–3), 146–162 (1954)
12. Jang, K.R., Myaeng, S.H.: Elucidating conceptual properties from word embeddings. In: SENSE 2017, pp. 91–96 (2017)
13. Jolliffe, I.: Principal component analysis. Wiley Online Library (2002)
14. Kutuzov, A., Andreev, I.: Texts in, meaning out: neural language models in semantic similarity tasks for Russian. *Komp'juternaja Lingvistika i Intellektual'nye Tehnologii* **2**(14), 133–144 (2015)
15. Kutuzov, A., Kuzmenko, E.: Comparing neural lexical models of a classic national corpus and a web corpus: the case for Russian. In: Gelbukh, A. (ed.) CICLing 2015. LNCS, vol. 9041, pp. 47–58. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18111-0_4
16. Kutuzov, A., Kuzmenko, E.: WebVectors: a toolkit for building web interfaces for vector semantic models. In: Ignatov, D.I., Khachay, M.Y., Labunets, V.G., Loukachevitch, N., Nikolenko, S.I., Panchenko, A., Savchenko, A.V., Vorontsov, K. (eds.) AIST 2016. CCIS, vol. 661, pp. 155–161. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-52920-2_15
17. Landauer, T.K., Dumais, S.T.: A solution to plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychol. Rev.* **104**(2), 211 (1997)
18. Lau, J.H., Newman, D., Baldwin, T.: Machine reading tea leaves: automatically evaluating topic coherence and topic model quality. In: EACL, pp. 530–539 (2014)
19. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Advances in Neural Information Processing Systems, pp. 2177–2185 (2014)
20. Levy, O., Goldberg, Y., Ramat-Gan, I.: Linguistic regularities in sparse and explicit word representations. In: CoNLL, pp. 171–180 (2014)

21. Luo, H., Liu, Z., Luan, H.B., Sun, M.: Online learning of interpretable word embeddings. In: EMNLP, pp. 1687–1692 (2015)
22. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. [arXiv:1301.3781](https://arxiv.org/abs/1301.3781) (2013)
23. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
24. Mu, J., Bhat, S., Viswanath, P.: All-but-the-top: simple and effective postprocessing for word representations. [arXiv:1702.01417](https://arxiv.org/abs/1702.01417) (2017)
25. Murphy, B., Talukdar, P.P., Mitchell, T.: Learning effective and interpretable semantic models using non-negative sparse embedding. In: COLING 2012 (2012)
26. Newman, D., Lau, J.H., Grieser, K., Baldwin, T.: Automatic evaluation of topic coherence. In: NACL, pp. 100–108. ACL (2010)
27. Nikolenko, S.I.: Topic quality metrics based on distributed word representations. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1029–1032. ACM (2016)
28. Panchenko, A., Loukachevitch, N.V., Ustalov, D., Paperno, D., Meyer, C.M., Konstantinova, N.: Russe: the first workshop on Russian semantic similarity. In: Dialogue, vol. 2, pp. 89–105 (2015)
29. Panchenko, A., Ustalov, D., Arefyev, N., Paperno, D., Konstantinova, N., Loukachevitch, N., Biemann, C.: Human and machine judgements for Russian semantic relatedness. In: Ignatov, D.I., Khachay, M.Y., Labunets, V.G., Loukachevitch, N., Nikolenko, S.I., Panchenko, A., Savchenko, A.V., Vorontsov, K. (eds.) AIST 2016. CCIS, vol. 661, pp. 221–235. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-52920-2_21
30. Ramrakhiyani, N., Pawar, S., Hingmire, S., Palshikar, G.K.: Measuring topic coherence through optimal word buckets. In: EACL 2017, pp. 437–442 (2017)
31. Rothe, S., Schütze, H.: Word embedding calculus in meaningful ultradense subspaces. In: Proceedings of ACL, p. 512 (2016)
32. Ruseti, S., Rebedea, T., Trausan-Matu, S.: Using embedding masks for word categorization. In: 1st Workshop on Representation Learning for NLP, pp. 201–205 (2016)
33. Smith, S.L., Turban, D.H., Hamblin, S., Hammerla, N.Y.: Offline bilingual word vectors, orthogonal transformations and the inverted softmax. [arXiv:1702.03859](https://arxiv.org/abs/1702.03859) (2017)
34. Tsvetkov, Y., Faruqui, M., Dyer, C.: Correlation-based intrinsic evaluation of word vector representations. In: 1st Workshop on Evaluating Vector Space Representations for NLP, pp. 111–115 (2016)

General Topics of Data Analysis

HuGaDB: Human Gait Database for Activity Recognition from Wearable Inertial Sensor Networks

Roman Chereshnev and Attila Kertész-Farkas^(✉)

National Research University Higher School of Economics (HSE),
Kochnovskiy Proezd, 3, Moscow 125319, Russia
akerteszfarkas@hse.ru

Abstract. This paper presents a human gait data collection for analysis and activity recognition consisting of continuous recordings of combined activities, such as walking, running, taking stairs up and down, sitting down, and so on; and the data recorded are segmented and annotated. Data were collected from a body sensor network consisting of six wearable inertial sensors (accelerometer and gyroscope) located on the right and left thighs, shins, and feet. Additionally, two electromyography sensors were used on the quadriceps (front thigh) to measure muscle activity. This database can be used not only for activity recognition but also for studying how activities are performed and how the parts of the legs move relative to each other. Therefore, the data can be used (a) to perform health-care-related studies, such as in walking rehabilitation or Parkinson's disease recognition, (b) in virtual reality and gaming for simulating humanoid motion, or (c) for humanoid robotics to model humanoid walking. This dataset is the first of its kind which provides data about human gait in great detail. The database is available free of charge <https://github.com/romanchereshnev/HuGaDB>.

1 Introduction

The increasing availability of wearable body sensors leads to novel scientific studies and industrial applications [1]. The main large areas include gesture recognition, human activity recognition, and human gait analysis. Several databases have been released for benchmarking; however, due to a wide variety of sensor types and the complexity of activities, these databases are rather distinct. Now, we will review these areas and the corresponding databases in a taxonomic manner.

Gesture recognition (GR) mainly focuses on recognizing hand-drawn gestures in the air. Patterns to be recognized may include numbers, circles, boxes, or Latin alphabet letters. Prediction is usually made on data obtained from smartphone sensors or some special gloves equipped with kinematic sensors, such as 3-axis accelerometers, 3-axis gyroscopes, and occasionally electromyography (EMG) sensors, to measure the electrical potential on the human skin during muscular activities [2]. A database for gesture recognition is available in [3].

Human activity recognition (HAR), on the other hand, aims at recognizing daily lifestyle activities. For instance, an interesting research topic is recognizing activities in or around the kitchen, such as cooking; loading the dishwasher or washing machine; preparing brownies or salads; scrambling eggs; light cleaning; opening or closing drawers, the fridge, or doors; and so on. Often these activities can be interrupted by, for example, answering phones. Databases on this topic include the MIT Place dataset [4,5], Darmstadt Daily Routine dataset [6], Ambient Kitchen [7], CMU Multi-Modal Activity Database (CMU-MMAC) [8], and Opportunity dataset [9,10]. In this topic, on-body inertial sensors are usually worn on the wrist, back, or ankle, however, additional sensors are used, such as temperature sensor, proximity sensor, water consumption sensor, heart rate and so on. For instance, CMU-MMAC includes videos, audios, RFID tags, motion capture system based on on-body markers, and physiological sensors such as galvanic skin response (GSR) and skin temperature, which are all located on both forearms and upper arms, left and right calves and thighs, abdomen, and wrists.

Other types of HAR usually focus on walking-related activities, such as walking, jogging, turning left or right, jumping, laying down, going up or down the stairs, and so on. Data on this topic can be found in the WARD dataset [11], PAMAP2 dataset [12,13], HASC challenge [14–16], USC-HAD [17,18], and MAREA [19]. For data collection, on-body sensors are often placed on the participant's wrist, waist, ankles, and back.

In some databases, exceptional efforts are taken to provide a reliable benchmark. The body sensor network conference (BSNC) (<http://bsncontest.org>) [20], for instance, has carried out a contest where organizers provided three different datasets from different research groups. Databases differ in sensor types used and activities recorded. Another team, called the Evaluating Ambient Assisted Living Systems through Competitive Benchmarking – Activity Recognition (EvAAL-AR), provides a service to evaluate HAR systems live on the same activity scenarios performed by an actor [21]. In this contest, each team brings its own activity recognition system, and the evaluation criteria attempt to capture the practical usability: recognition accuracy, user acceptance, recognition delay, installation complexity, and interoperability with ambient-assisted living systems.

Gait analysis focuses not only on the recognition of activities observed but also on how activities are performed. This can be useful in health-care systems for monitoring patients recovering after surgery or fall detection or in diagnosing the state of, for example, Parkinson's disease [22,23]. For instance, the Daphnet Gait dataset (DG) [24] consists of recordings of 10 participants affected with Parkinson's disease instructed to carry out activities that are likely to be difficult to perform, such as walking. The objective is to detect these incidents from accelerometer data recorded from above the ankle, above the knee, and on the trunk. On the other hand, Bovi et al. provide a gait dataset collected from 40 healthy people with various ages as a reference dataset [25]. In the aforementioned BSNC, the third database (ID:IC) contains gait data before knee surgery and 1, 3, 6, 12, and 24 weeks (respectively) after it.

2 Motivation and Design Goals

The main purpose of this dataset is to provide detailed gait data to study how the parts of the legs move individually and relative to each other during activities such as walking, running, standing up, and so on. A summary of the activities can be found in Table 1. This dataset contains continuous recordings of combinations of activities, and the data are segmented and annotated with the label of the activity currently performed. Thus, this dataset is also suitable for analyzing human gait and activities between transitions.

Table 1. Characteristics of HuGaDB

ID	Activity	Time sec (min)	Percent	Samples	Description
1	Walking	11544 (192)	32.15	679073	Walking and turning at various speeds on a flat surface
2	Running	1218 (20)	3.39	71653	Running at various paces
3	Going up	2237 (37)	6.23	131604	Taking stairs up at various speeds
4	Going down	1982 (33)	5.52	116637	Taking the stairs down at various speeds and steps
5	Sitting	4111 (68)	11.45	241849	Sitting on a chair; sitting on the floor not included
6	Sitting down	409 (6)	1.14	24112	Sitting on a chair; sitting down on the floor not included
7	Standing up	380 (6)	1.06	22373	Standing up from a chair
8	Standing	5587 (93)	15.56	328655	Static standing on a solid surface
9	Bicycling	2661 (44)	7.41	156560	Typical bicycling
10	Up by elevator	1515 (25)	4.22	89144	Standing in an elevator while moving up
11	Down by elevator	1185 (19)	3.30	69729	Standing in an elevator while moving down
12	Sitting in car	3069 (51)	8.55	180573	Sitting while travelling by car as a passenger
	Total	35903	598	2111962	

Mainly inertial sensors were used for data acquisition. We decided to use inertial sensors because they are inexpensive, simple to use anywhere such as indoor and outdoor area, and widely available compared with other systems. For instance, compared with video-based motion capture systems, they require expensive video cameras and special full bodysuit with special markers on it. In addition, they are restricted to being used in the installed test area and they are sensitive to lightning and suffer from lost markers phenomenon.

In total, six inertial sensors were placed on the right and left thighs, shins and feet; and data were collected from 18 healthy participants, providing total 10 h of recording. This allows one to investigate how the parts of the legs move individually and relative to each other within and in-between activities. Our dataset could be used as control data, for instance, in health-care-related studies, such as walking rehabilitation or Parkinson's disease recognition. In virtual reality or gaming, our dataset can be used to model a virtual human movements by reproducing the leg movements from the accelerometer data by simply taking the integrals. In fact, it is not limited to virtual environment and could be used to train to walk and move humanoid robots to make them more humanlike and cope with the uncanny valley.

This dataset is unique in the sense that it is the first to provide human gait data in great detail mainly from inertial sensors and contains segmented annotations for studying the transition between different activities.

3 Data Collection and Sensor Network Topology

In data collection, we used MPU9250 inertial sensors and electromyography (EMG) sensors. Each EMG sensor has a voltage gain is about 5000 and band-pass filter with bandwidth corresponding to power spectrum of EMG (10–500 Hz). A sample rate of each EMG-channel is 1.0 kHz, ADC resolution is 8 bits, input voltages: 0–5 V. The inertial sensors consisted of a 3-axis accelerometer and a 3-axis gyroscope integrated into a single chip. Data were collected with accelerometer's range equal to ± 2 g with sensitivity 16.384 LSB/g and gyroscope's range equal to $\pm 2000^\circ/\text{s}$ with sensitivity 16.4 LSB $^\circ/\text{s}$. All sensors are powered from a battery, that helps to minimize electrical grid noise.

Accelerometer and gyroscope signals were stored in int16 format. EMG signals are stored in uint8. Therefore, accelerometer data can be converted to m/s^2 by dividing raw data 32768 and multiplying it by 2g. Raw gyroscope data can be converted to $^\circ/\text{s}$ by multiplying it by 2000/32768. Raw EMG data can be converted to Volts by multiplying it 0.001/255. We kept the raw data in our data collection in case one prefers other normalization techniques.

In total, three pairs of inertial sensors and one pair of EMG sensors were installed symmetrically on the right and left legs with elastic bands. A pair of inertial sensors were installed on the rectus femoris muscle 5 cm above the knee, a pair of sensors around the middle of the shinbone at the level where the calf ends, and a pair on the feet on the metatarsal bones. Two EMG sensors were placed on vastus lateralis and connected to the skin with three electrodes. The locations of the sensors are shown in Fig. 1. In total, 38 signals were collected, 36 from the inertial sensors and 2 from the EMG sensors.

The sensors were connected through wires with each other and to a microcontroller box, which contained an Arduino electronics platform with a Bluetooth module. The microcontroller collected 56.3500 samples per second in average with standard deviation (std) 3.2057 and then transmitted them to a laptop through Bluetooth connection.

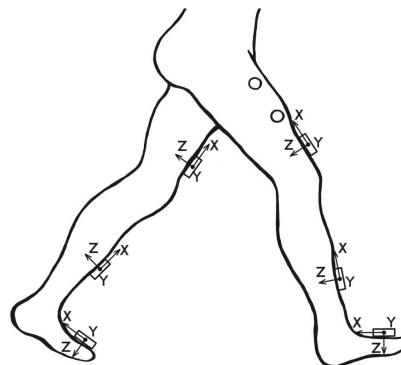


Fig. 1. Location of sensors. EMG sensor are shown as circles while boxes represent inertial sensors

The data were collected from 18 participants. These participants were healthy young adults: 4 females and 14 males, average age of 23.67 (std: 3.69) years, an average height of 179.06 (std: 9.85) cm, and an average weight of 73.44 (std: 16.67) kg.

The participants performed a combination of activities at normal speed and casual way, and there were no obstacles placed on their way. For instance, a participant was instructed to perform the following activities: starting from a sitting position, sitting - standing up - walking - going up the stairs - walking - sitting down. The experimenter recorded the data continually using a laptop and annotated the data with the activities performed. This provided us a long, continuous sequence of segmented data annotated with activities. We developed our own data collector program. In total, 2,111,962 samples were collected from all the 18 participants, and they provided a total of 10 h of data.

Data acquisition was carried out mainly inside a building. However, activities such as running, bicycling, and sitting in a car were performed outside. We collected data in a moving elevator and vehicle. In these scenarios, the activities performed were simply standing or sitting. However, a force impact on the accelerometer sensors and in certain applications, it may be important to consider these facts. Note that we did not collect data on a treadmill.

4 Data Format

Data obtained from the sensors were stored in flat text files. We decided to store the data in flat files because they have one of the most universal formats, and they can be easily preprocessed in all programming languages on every system. One data file contains one recording, which is either a single activity (e.g., walking) or a series of activities. Every file name was created according to the template **HGD_vX_ACT_PR_CNT.txt**. HGD is a prefix that means human gait data and vX means the version of the data files, currently v1. ACT is a variable, and it

denotes the activity ID that was performed. If a file contains a series of different types of activities, then it is indicated as VARIOUS. PR indicates the ID of the person who performed the activity. Data recording was repeated a few times, and CNT is a counter for this. For example, a file named HGD_v1_walking_17_02.txt contains data from participant 17 while he was walking for the second time. The file naming convention is summarized in Table 2.

Table 2. Description of the file naming convention

TAG	Description	Type	Comment
HGD	Prefix	Fixed	Data files start with this prefix
vX	Version number	Integer	Indicates the version of the data format
ACT	Activity	String	Indicates the type of activity
PR	Participant ID	Integer	Indicates the subject whose data was recorded
CNT	Counter	Integer	Counter for repeated experiments

The main body of the data files contains tab-delimited raw, unnormalized data obtained from the sensors directly. Each data file starts with a header, which contains metainformation. It summarizes the list of activities, the IDs of the activities recorded, and the time and date of the recording. This is summarized in Table 3.

Table 3. Description of the data file header

TAG	Description	Type	Comment
#Activity	List of the activities	String	Lists the activity types in this file
#ActivityID	List of the ID of activities	List of integers	Lists the activity types in this file
#Date-Time	Date and Time	YEAR-MM-DD-HR-MN	Year-Month-Day-Hour-Min format

The main data body of every file has 39 columns. Each column corresponds to a sensor, and one row corresponds to a sample. The order of the columns is fixed. The first 36 columns correspond to the inertial sensors, the next 2 columns correspond to the EMG sensors, and the last column contains the activity ID. The activities are coded as shown in Table 1. The inertial sensors are listed in the following order: right foot (RF), right shin (RS), right thigh (RT), left foot (LT), left shin (LS), and left thigh (LT), followed by right EMG (R) and left EMG (L). Each inertial sensor produces three acceleration data on x, y, z axes and three gyroscope data on x, y, z axes. For instance, the column named ‘RT_acc_z’ contains data obtained from the z-axis of accelerometer located on the right thigh.

Sample data with respect to the activities are visualized through a heat map representation in Fig. 2.

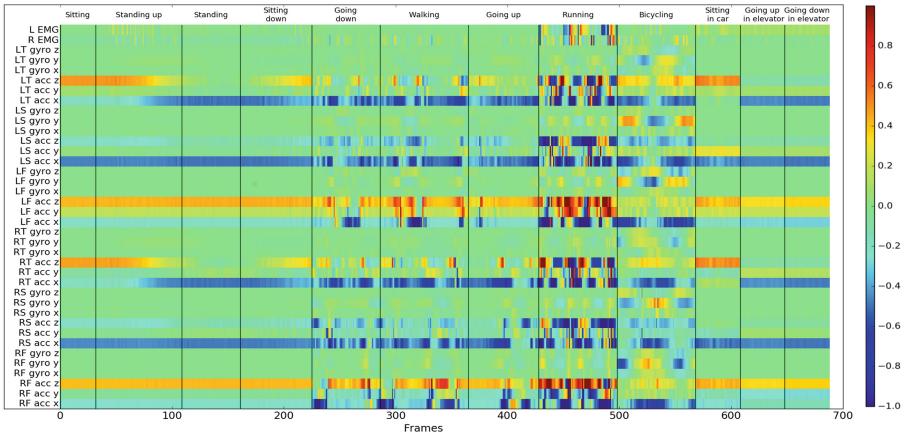


Fig. 2. Data visualization. For normalization data from initial sensors were divided by 32768 and data from EMG were subtracted by 128 and divided by 128

A screenshot of some part of data file can be seen in Fig. 3

```
#ActivityID 0 5 7 6 5 7 8 6 5 7 8 6 5
#Date 10-26-13 47:47:00
#Sensors L_EMG R_EMG LT_gyro_z LT_gyro_x LT_acc_z LT_acc_x LS_gyro_z LS_gyro_y LS_gyro_x LS_acc_z LS_acc_y LS_acc_x LF_gyro_z LF_gyro_y LF_gyro_x LF_acc_z LF_acc_x LF_acc_y RT_gyro_z RT_gyro_x RT_acc_z RT_acc_x RT_acc_y RS_gyro_z RS_gyro_x RS_acc_z RS_acc_x RF_gyro_z RF_gyro_x RF_acc_z RF_acc_x RF_acc_y

-7654 -1200 13444 16 11 -22 -14464 520 -7450 -6 -2 22 -6528 -1016 13256 38 -7 -6 -6912 5720 13824 38 -7 -22 -15176 408 -7168 12 -24 38 -6064 968 1390 37 -10 -21 120 1
-7712 -1094 13444 16 11 -22 -14464 520 -7450 -6 -2 22 -6528 -1016 13256 38 -7 -6 -6912 5720 13824 38 -7 -22 -15176 408 -7168 12 -24 38 -6064 968 1390 37 -10 -21 120 1
-7722 -1094 13446 23 8 -20 -14767 592 -7264 25 6 29 -6600 -832 13232 50 -9 -14 -7088 5704 14032 40 -13 -18 -15080 568 -6920 34 -21 48 -5720 904 16120 45 -14 -35 120 1
-7692 -1094 13446 20 10 -23 -14460 624 -7368 33 2 33 -6408 -832 13236 44 -6 -21 -7008 5768 13800 47 -3 -19 -16166 560 -6920 36 -25 50 -6362 1024 14168 44 -18 -38 1
-7702 -1094 13446 20 10 -23 -14460 624 -7368 33 2 33 -6408 -832 13236 44 -6 -21 -7008 5768 13800 47 -3 -19 -16166 560 -6920 36 -25 50 -6362 1024 14168 44 -18 -38 1
-7708 -1048 13495 10 11 -24 -14416 864 -7760 23 0 27 -6616 -656 13446 38 -14 -8 -7064 5840 13972 40 -11 -19 -15120 608 -7056 12 -19 47 -5632 1248 1423 62 -12 -23 1
-7692 -1094 13495 10 11 -24 -14416 864 -7760 23 0 27 -6616 -656 13446 38 -14 -8 -7064 5840 13972 40 -11 -19 -15120 608 -7056 12 -19 47 -5632 1248 1423 62 -12 -23 1
-7692 -1094 13592 -6 -24 -14394 712 -7468 -18 -6 13 -6760 -576 13216 29 -10 15 -6928 5762 13760 49 -17 -15140 608 -7056 12 -19 47 -5632 1248 1423 62 -12 -23 1
-7692 -1094 13592 -6 -24 -14394 712 -7468 -18 -6 13 -6760 -576 13216 29 -10 15 -6928 5762 13760 49 -17 -15140 608 -7056 12 -19 47 -5632 1248 1423 62 -12 -23 1
-7648 -9364 13604 -9 -17 -22 -14468 824 -7548 -11 -4 -6424 -436 13200 18 -16 24 -7048 5748 13872 42 -2 -19 -15176 704 -7056 18 -14 48 -5592 1382 14152 63 -8 -14 1
-7658 -1000 13800 -14 14 -22 -14424 624 -7480 -16 -10 -6 -6882 -456 13844 9 -28 21 -6992 5864 13936 49 -13 -11 -15140 608 -7056 12 -19 47 -5632 1248 1423 62 -12 -23 1
-7688 -1000 13800 -14 14 -22 -14424 624 -7480 -16 -10 -6 -6882 -456 13844 9 -28 21 -6992 5864 13936 49 -13 -11 -15140 608 -7056 12 -19 47 -5632 1248 1423 62 -12 -23 1
-7648 -1000 13874 -19 20 -27 -14460 640 -7776 7 -10 -19 -6688 -468 13160 6 -20 22 -7096 5892 13898 36 -12 -21 -15120 792 -7112 18 -11 46 -5689 1456 15992 60 -9 -1
-7722 -1094 13874 -19 20 -27 -14460 640 -7776 7 -10 -19 -6688 -468 13160 6 -20 22 -7096 5892 13898 36 -12 -21 -15120 792 -7112 18 -11 46 -5689 1456 15992 60 -9 -1
-7692 -1094 13800 -21 30 -28 -14504 704 -7712 -16 -17 -21 -6720 -696 13200 5 -28 -7048 5888 13768 27 -2 -26 -15296 720 -7112 18 -10 29 -5672 1528 16128 44 -9 -10 1
-7722 -1094 13800 -21 30 -28 -14504 704 -7712 -16 -17 -21 -6720 -696 13200 5 -28 -7048 5888 13768 27 -2 -26 -15296 720 -7112 18 -10 29 -5672 1528 16128 44 -9 -10 1
-7696 -1016 13500 -21 30 -28 -14504 704 -7712 -16 -17 -21 -6720 -696 13200 5 -28 -7048 5888 13768 27 -2 -26 -15296 720 -7112 18 -10 29 -5672 1528 16128 44 -9 -10 1
-7736 -1032 13480 -28 26 -31 -14424 582 -7712 -7 -19 -24 -6654 -584 13144 -8 -28 29 -7096 5888 13472 14 -16 -22 -15154 832 -7264 3 -12 18 -5712 1744 16032 39 -12 -4 1
-7736 -1040 13480 -28 26 -31 -14424 582 -7712 -7 -19 -24 -6654 -584 13144 -8 -28 29 -7096 5888 13472 14 -16 -22 -15154 832 -7264 3 -12 18 -5712 1744 16032 39 -12 -4 1
-7716 -1120 13532 -35 31 -31 -14464 560 -7456 -3 -27 -36 -6568 -504 13224 -16 -29 36 -6960 5864 13728 1 -9 -22 -15152 782 -6952 4 -6 -2 -5576 1288 16248 2 3 21 1
```

Fig. 3. Screenshot of the data file

The data files can be loaded easily in most of the popular programming languages. For instance, they can be loaded in Python using the following script:

```
import numpy as np
data = np.genfromtxt(path_to_file,
                     delimiter='\t',
                     skip_header=4)
```

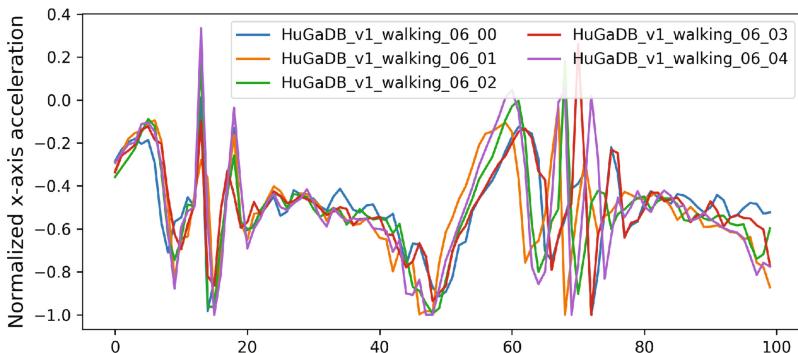
Please note that it requires NumPy library. It also can be loaded in Matlab with the following one-line command:

```
data = dlmread(path_to_file, '\t', 4, 0);
```

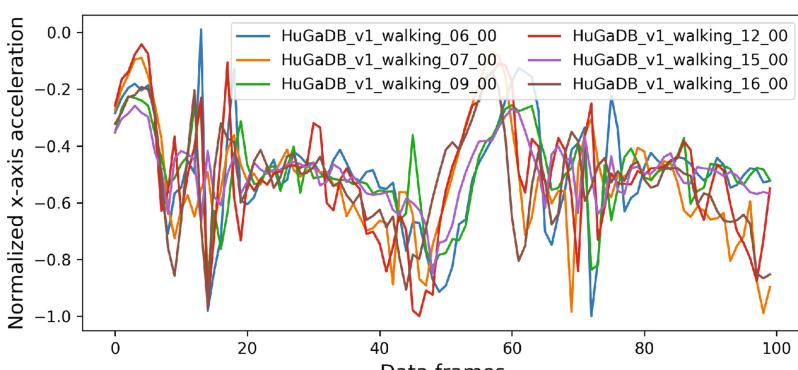
We have prepared a script to load the data into SQLite database, which is available at the database's website: https://github.com/romanchereshnev/HuGaDB/blob/master/Scripts/create_db.py.

5 Discussion on Data Variance

We were interested seeing the variance among the data collected, in particular, the data variance (A) within a single user and (B) between several users. For this reason, we plotted in Fig. 4 the x-axes acceleration data from the thigh recorded during a short two-three-step walk. Panel A shows the data from various recordings performed by the same user. It can be seen that the data variance at a single



(A) Single user



(B) Various users

Fig. 4. Data variance during walking. (A) Activity performed by the same user multiple times. (B) Activity performed by different users. Legend indicates the source of the data. Data are scaled to the range $[-1, +1]$.

frame is quite low suggesting that people perform activities very similar way. On the other hand, panel B shows data obtained from six different, randomly chosen users. Here, a much higher variance can be seen in the same frames compared to the previous case. The increased variance may arise from several facts including: difference in gait, difference in leg shape, sensors mounted in slightly different positions, etc. We obtained similar conclusions on data obtained from different sensors during different activities. We note that, even higher variance was observed in the EMG data, which resulted from the difference in the electricity conduction characteristics of the skin, skin thickness, etc.

Taking into account the high data variance between different users, we emphasize the importance of proper evaluation of machine learning methods developed for human activity recognition. Therefore, we propose using the supervised cross-validation approach for constructing training and test sets [26]. In this approach, all the data from a designated user are held out only for tests and the data from the other 17 participants are used for training. Thus, this approach provides a reliable estimation of how an activity recognition system would perform with a new user whose data was not seen before.

Variance can arise from using different brands of sensors. Unfortunately, we did not have the capacity to collect data from different brands of sensors. We hope the measurement noise is small in general and that different sensors can be calibrated to be compatible with each other.

6 Availability

The database is available free of charge at <https://github.com/romanchereshnev/HuGaDB> (455 Mb).

7 Summary

The HuGaDB dataset contains detailed kinematic data for analyzing human gait and activity recognition. This dataset differs from previously published datasets in the sense that HuGaDB provides human gait data in great detail mainly from inertial sensors and contains segmented annotations for studying the transition between different activities. Data were obtained from 18 participants, and in total, they provide around 10 h of recording. This dataset can be used in health-care-related studies, such as walking rehabilitation, or in modeling human movements in virtual reality or humanoid robotics. The dataset will be updated with new data from new participants in the future.

References

1. Aggarwal, C.C.: Managing and Mining Sensor Data. Springer Science & Business Media, New York (2013). <https://doi.org/10.1007/978-1-4614-6309-2>
2. Amma, C., Georgi, M., Schultz, T.: Airwriting: a wearable handwriting recognition system. Pers. Ubiquit. Comput. **18**(1), 191–203 (2014)

3. Georgi, M., Amma, C., Schultz, T.: Recognizing hand and finger gestures with IMU based motion and EMG based muscle activity sensing. In: Proceedings of the International Conference on Bio-inspired Systems and Signal Processing, pp. 99–108 (2015)
4. Tapia, E.M., Intille, S.S., Lopez, L., Larson, K.: The design of a portable kit of wireless sensors for naturalistic data collection. In: Fishkin, K.P., Schiele, B., Nixon, P., Quigley, A. (eds.) Pervasive 2006. LNCS, vol. 3968, pp. 117–134. Springer, Heidelberg (2006). https://doi.org/10.1007/11748625_8
5. Intille, S.S., Larson, K., Beaudin, J., Nawyn, J., Tapia, E.M., Kaushik, P.: A living laboratory for the design and evaluation of ubiquitous computing technologies. In: CHI 2005 Extended Abstracts on Human Factors in Computing Systems, pp. 1941–1944. ACM (2005)
6. Huynh, T., Fritz, M., Schiele, B.: Discovery of activity patterns using topic models. In: Proceedings of the 10th International Conference on Ubiquitous Computing, pp. 10–19. ACM (2008)
7. Pham, C., Olivier, P.: Slice&Dice: recognizing food preparation activities using embedded accelerometers. In: Tscheligi, M., et al. (eds.) European Conference on Ambient Intelligence, AmI 2009. LNCS, vol. 5859, pp. 34–43. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-05408-2_4
8. De la Torre, F., Hodgins, J., Bargteil, A., Martin, X., Macey, J., Collado, A., Beltran, P.: Guide to the Carnegie Mellon University multimodal activity (CMU-MMAC) database, p. 135. Robotics Institute (2008)
9. Chavarriaga, R., Sagha, H., Calatroni, A., Digumarti, S.T., Tröster, G., del R. Millán, J., Roggen, D.: The opportunity challenge: a benchmark database for on-body sensor-based activity recognition. Pattern Recogn. Lett. (2013)
10. Sagha, H., Digumarti, S.T., Millán, J.d.R., Chavarriaga, R., Calatroni, A., Roggen, D., Tröster, G.: Benchmarking classification techniques using the opportunity human activity dataset. In: 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 36–40. IEEE (2011)
11. Yang, A.Y., Kuryloski, P., Bajcsy, R.: WARD: a wearable action recognition database (2009)
12. Reiss, A., Stricker, D.: Introducing a new benchmarked dataset for activity monitoring. In: 2012 16th International Symposium on Wearable Computers, pp. 108–109. IEEE (2012)
13. Reiss, A., Stricker, D.: Creating and benchmarking a new dataset for physical activity monitoring. In: Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments, 40 pages. ACM (2012)
14. Kawaguchi, N., Ogawa, N., Iwasaki, Y., Kaji, K., Terada, T., Murao, K., Inoue, S., Kawahara, Y., Sumi, Y., Nishio, N.: HASC challenge: gathering large scale human activity corpus for the real-world activity understandings. In: Proceedings of the 2nd Augmented Human International Conference, 27 pages. ACM (2011)
15. Kawaguchi, N., Watanabe, H., Yang, T., Ogawa, N., Iwasaki, Y., Kaji, K., Terada, T., Murao, K., Hada, H., Inoue, S., et al.: HASC2012corpus: large scale human activity corpus and its application. In: Proceedings of the IPSN, vol. 12 (2012)
16. Kawaguchi, N., Yang, Y., Yang, T., Ogawa, N., Iwasaki, Y., Kaji, K., Terada, T., Murao, K., Inoue, S., Kawahara, Y., et al.: HASC2011corpus: towards the common ground of human activity recognition. In: Proceedings of the 13th International Conference on Ubiquitous Computing, pp. 571–572. ACM (2011)
17. Zhang, M., Sawchuk, A.A.: USC-HAD: a daily activity dataset for ubiquitous activity recognition using wearable sensors. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, pp. 1036–1043. ACM (2012)

18. Zhang, M., Sawchuk, A.A.: Human daily activity recognition with sparse representation using wearable sensors. *IEEE J. Biomed. Health Inform.* **17**(3), 553–560 (2013)
19. Khandelwal, S., Wickström, N.: Evaluation of the performance of accelerometer-based gait event detection algorithms in different real-world scenarios using the Marea gait database. *Gait Posture* **51**, 84–90 (2017)
20. Giuberti, M., Ferrari, G.: Simple and robust BSN-based activity classification: winning the first BSN contest. In: Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies, 34 pages. ACM (2011)
21. Gjoreski, H., Kozina, S., Gams, M., Lustrek, M., Álvarez-García, J.A., Hong, J.H., Dey, A.K., Bocca, M., Patwari, N.: Competitive live evaluations of activity-recognition systems. *IEEE Pervasive Comput.* **14**(1), 70–77 (2015)
22. Sant'Anna, A., Salarian, A., Wickstrom, N.: A new measure of movement symmetry in early Parkinson's disease patients using symbolic processing of inertial sensor data. *IEEE Trans. Biomed. Eng.* **58**(7), 2127–2135 (2011)
23. Sant'Anna, A.: A symbolic approach to human motion analysis using inertial sensors: framework and gait analysis study. Ph.D. thesis, Halmstad University (2012)
24. Bachlin, M., Roggen, D., Troster, G., Plotnik, M., Inbar, N., Meidan, I., Herman, T., Brozgol, M., Shaviv, E., Giladi, N., et al.: Potentials of enhanced context awareness in wearable assistants for Parkinson's disease patients with the freezing of gait syndrome. In: 2009 International Symposium on Wearable Computers, pp. 123–130. IEEE (2009)
25. Bovi, G., Rabuffetti, M., Mazzoleni, P., Ferrarin, M.: A multiple-task gait analysis approach: kinematic, kinetic and emg reference data for healthy young and adult subjects. *Gait Posture* **33**(1), 6–13 (2011)
26. Kertész-Farkas, A., Dhir, S., Sonego, P., Pacurar, M., Netoteia, S., Nijveen, H., Kuzniar, A., Leunissen, J.A., Kocsor, A., Pongor, S.: Benchmarking protein classification algorithms via supervised cross-validation. *J. Biochem. Biophys. Methods* **70**(6), 1215–1223 (2008)

On Finding Maximum Cardinality Subset of Vectors with a Constraint on Normalized Squared Length of Vectors Sum

Anton V. Eremeev^{1,2(✉)}, Alexander V. Kelmanov^{3,4}, Artem V. Pyatkin^{3,4},
and Igor A. Ziegler^{1,2}

¹ Omsk Branch of Sobolev Institute of Mathematics SB RAS, Omsk, Russia
eremeev@ofim.oscsbras.ru

² Omsk State University n.a. F.M. Dostoevsky, Omsk, Russia
ziegler.igor@gmail.com

³ Sobolev Institute of Mathematics SB RAS, Novosibirsk, Russia
{kelm,artem}@math.nsc.ru

⁴ Novosibirsk State University, Novosibirsk, Russia

Abstract. In this paper, we consider the problem of finding a maximum cardinality subset of vectors, given a constraint on the normalized squared length of vectors sum. This problem is closely related to Problem 1 from (Eremeev, Kel'manov, Pyatkin, 2016). The main difference consists in swapping the constraint with the optimization criterion.

We prove that the problem is NP-hard even in terms of finding a feasible solution. An exact algorithm for solving this problem is proposed. The algorithm has a pseudo-polynomial time complexity in the special case of the problem, where the dimension of the space is bounded from above by a constant and the input data are integer. A computational experiment is carried out, where the proposed algorithm is compared to COINBONMIN solver, applied to a mixed integer quadratic programming formulation of the problem. The results of the experiment indicate superiority of the proposed algorithm when the dimension of Euclidean space is low, while the COINBONMIN has an advantage for larger dimensions.

Keywords: Vectors sum · Subset selection · Euclidean norm
NP-hardness · Pseudo-polynomial time

1 Introduction

In this paper, we study a discrete extremal problem of searching a subset of vectors with maximum cardinality, given a constraint on the normalized squared length of vectors sum. The main goal of the study is to test experimentally two different approaches to solving this problem. The first approach is based on the dynamic programming and the second one is based on the mixed-integer mathematical programming. We also comment on the computational complexity of this problem and estimate the time complexity of a proposed algorithm based on the dynamic programming principles.

The Maximum Cardinality Subset of Vectors with a Constraint on Normalized Squared Length of Vectors Sum (MCSV) problem is formulated as follows.

Given: a set $\mathcal{Y} = \{y_1, \dots, y_N\}$ of points (vectors) from \mathbb{R}^q and a number $\alpha \in (0, 1)$.

Find: a subset $\mathcal{C} \subseteq \mathcal{Y}$ of maximum cardinality such that

$$\frac{1}{|\mathcal{C}|} \left\| \sum_{y \in \mathcal{C}} y \right\|^2 \leq \alpha \frac{1}{|\mathcal{Y}|} \left\| \sum_{y \in \mathcal{Y}} y \right\|^2, \quad (1)$$

where $\|\cdot\|$ denotes the Euclidean norm.

If the given points of the Euclidean space correspond to people so that the coordinates of points are equal to some characteristics of these people, then the MCSV problem may be treated as a problem of finding a sufficiently balanced group of people of maximum size.

MCSV problem is closely related to Problem 1 from [5]. The main difference consists in swapping the constraint with the optimization criterion. The problems of finding a subset of vectors, analogous to the MCSV problem are typical in the Data editing and Data cleaning, where one needs to exclude some error observations from the sample (see e.g. [7, 9, 10]). A recent example of such a problem may be found in [1], where a maximum cardinality subset of vectors is sought, given a constraint that a quadratic spread of points in the subset w.r.t. its centroid is upper-bounded by a pre-specified portion of the total quadratic spread of points in the input set w.r.t. the centroid of that set.

To compare the MCSV to the problem considered in [1], we note that

$$\frac{1}{|\mathcal{C}|} \left\| \sum_{y \in \mathcal{C}} y \right\|^2 = \sum_{y \in \mathcal{C}} \|y\|^2 - \sum_{y \in \mathcal{C}} \|y - \bar{y}(\mathcal{C})\|^2.$$

In the right-hand side, the first sum is the total quadratic spread of points with respect to zero, the second one is relative to the centroid $\bar{y}(\mathcal{C})$ of \mathcal{C} . The value $\frac{1}{|\mathcal{Y}|} \left\| \sum_{y \in \mathcal{Y}} y \right\|^2 = \sum_{y \in \mathcal{Y}} \|y\|^2 - \sum_{y \in \mathcal{Y}} \|y - \bar{y}(\mathcal{Y})\|^2$ characterizes the difference of analogous quadratic spreads in the initial set. Therefore the MCSV problem asks for a subset of maximum size such that, in this subset, the two mentioned above total quadratic spreads differ by not more than α times from the same difference in the input set \mathcal{Y} .

The MCSV problem may be also treated as a Boolean optimization problem with a quadratic constraint:

$$\sum_{i=1}^N x_i \rightarrow \max, \quad (2)$$

s.t.

$$\sum_{j=1}^q \left(\sum_{i=1}^N y_i^{(j)} x_i \right)^2 \leq \alpha \frac{1}{N} \sum_{j=1}^q \left(\sum_{i=1}^N y_i^{(j)} \right)^2 \cdot \sum_{i=1}^N x_i, \quad (3)$$

$$x_i \in \{0, 1\}, \quad i = 1, \dots, N, \quad (4)$$

where

- N is the cardinality of set \mathcal{Y} ,
- q is the dimension of the Euclidean space,
- $y_i^{(j)}$ is j -th coordinate of the i -th vector,
- x_i is a Boolean variable, $x_i = 1$ if the i -th vector is included in the solution; otherwise $x_i = 0$ ($i = 1, \dots, N$).

Another problem related to the MCSV is the trading hubs construction problem, emerging in electricity markets under locational marginal pricing [2–4]. A trading hub is a subset of nodes of the electricity grid that may be used to calculate a *price index* as an average nodal price over the hub nodes. This price index may be employed by the market participants for hedging by the means of futures contacts [2]. Assume that the set of nodes of the electricity grid which may be included into a hub is $\{1, \dots, N\}$ and c_{it} is the price at node i , $i = 1, \dots, N$, at an hour t , $t = 1, \dots, T$, where T is the length of a historic period for which the electricity prices are observed. Let p_{rt} denote the electricity price of participant r , $r = 1, \dots, R$, at hour t , and R is the number of participants. The single hub construction problem consists in minimizing the sum of squared differences of the prices of participants from the hub price, requiring that the hub contains at least n_{\min} nodes:

$$\text{Min} \sum_{t=1}^T \sum_{r=1}^R (c_t - p_{rt})^2 \quad (5)$$

s.t.

$$c_t = \frac{\sum_{i=1}^N x_i c_{it}}{\sum_{i=1}^N x_i}, \quad t = 1, \dots, T, \quad (6)$$

$$\sum_{i=1}^N x_i \geq n_{\min}, \quad (7)$$

$$x_i \in \{0, 1\}, \quad i = 1, \dots, N, \quad c_t \geq 0, \quad t = 1, \dots, T. \quad (8)$$

Here the binary variables x_i turn into 1 whenever node i is included into the hub. The variables c_t define the hub price at time t , $t = 1, \dots, T$. This problem is proved to be NP-hard in [2], where a genetic algorithm was proposed for finding approximate solutions to it. In the special case of a single market participant problem (5)–(7) is equivalent to the MCSV problem, where the criterion (5) and constraint (7) are swapped and instead of a lower bound on the hub size (which turns into a maximization criterion in MCSV) we are given an upper bound on the sum of squared differences of the prices of participants from the hub price. Such a modification of single hub construction problem may be appropriate in situations where the required closeness of a hub price to the prices of participants may be defined, e.g. on the basis of an observation an already existing hub [8].

The paper has the following structure. In Sect. 2, we show that MCSV problem is NP-hard even in terms of finding a feasible solution. An exact algorithm for solving this problem using the dynamic programming approach is proposed in Sect. 3. The algorithm has a pseudo-polynomial time complexity in the special case of the problem, where the dimension q of the space is bounded from

above by a constant and the input data are integer. Section 4 describes how the algorithm is implemented and a computational experiment is carried out. The purpose of the experiment is to analyze the algorithm and compare it with the COINBONMIN solver.

2 Problem Complexity

The following proposition shows that MCSV problem is NP-hard even in terms of finding a feasible solution.

Theorem 1. *Finding out whether MCSV has a feasible solution is NP-hard.*

Proof. Consider an instance of the Exact Cover by 3-sets problem, i. e. the family of subsets A_1, \dots, A_n of a set A where $|A_i| = 3$ for all i and $|A| = 3p$ where p is an integer. The question is whether there are p subsets in this family whose union is A . This problem is known to be NP-complete [6].

Put $q = 3p$, $N = n + 1$ and for each $i = 1, \dots, n$ let y_i be a characteristic vector of the set A_i (i. e. the j -th coordinate of y_i is 1 if $j \in A_i$ and 0 otherwise). Put $y_N = (-1, \dots, -1)$ and choose α in such a way that $\alpha \|\sum_{y \in \mathcal{Y}} y\|^2 < 3$. Then the constructed instance has a feasible solution if and only if there is an exact cover by 3-sets. Indeed, if there is no such cover then for each non-empty set \mathcal{C} we have

$$\frac{1}{|\mathcal{C}|} \left\| \sum_{y \in \mathcal{C}} y \right\|^2 \geq \frac{3}{N} > \frac{\alpha}{|\mathcal{Y}|} \left\| \sum_{y \in \mathcal{Y}} y \right\|^2$$

by the choice of α , i. e. there are no feasible solutions. The opposite implication is trivial.

3 A Pseudo-polynomial Time Algorithm for Bounded Dimension of Space

In this section, we show that in the case of a fixed dimension q of the space and integer coordinates of vectors from \mathcal{Y} , the MCSV problem can be solved in a pseudo-polynomial time using the same approach as proposed in [5].

For arbitrary sets $\mathcal{P}, \mathcal{Q} \subset \mathbb{R}^q$ define their sum as

$$\mathcal{P} + \mathcal{Q} = \{x \in \mathbb{R}^q \mid x = y + y', \ y \in \mathcal{P}, \ y' \in \mathcal{Q}\}. \quad (9)$$

For every positive integer r denote by $\mathcal{B}(r)$ the set of all vectors in \mathbb{R}^q whose coordinates are integer and at most r by absolute value. Then $|\mathcal{B}(r)| \leq (2r+1)^q$.

Let b be the maximum absolute value of all coordinates of the input vectors y_1, \dots, y_N . Our algorithm for the MCSV problem successively computes the subsets $\mathcal{S}_k \subseteq \mathcal{B}(bk)$, $k = 1, \dots, N$, where each subset \mathcal{S}_k contains all vectors that can be obtained by summing different elements of the set $\{y_1, \dots, y_k\}$.

For $k = 1$ we assume $\mathcal{S}_1 = \{\mathbf{0}, y_1\}$. Then we compute

$$\mathcal{S}_k = \mathcal{S}_{k-1} + (\{\mathbf{0}\} \cup \{y_k\})$$

for all $k = 2, \dots, N$, using the formula (9).

For each element $z \in \mathcal{S}_k$ we store an integer parameter n_z and a subset $\mathcal{C}_z \subseteq \mathcal{Y}$ such that $z = \sum_{y \in \mathcal{C}_z} y$, where $|\mathcal{C}_z| = n_z$ and n_z is the maximum number of addends that were used to produce z .

When the subset \mathcal{S}_N is computed, we find an element $z^* \in \mathcal{S}_N$ such that $\|z^*\|^2/n_{z^*} \leq \alpha \frac{1}{|\mathcal{Y}|} \|\sum_{y \in \mathcal{Y}} y\|^2$ and the value n_{z^*} is maximum (if such elements exist in \mathcal{S}_N). The result of the algorithm is the subset \mathcal{C}_{z^*} corresponding to the found vector z^* or a conclusion that the problem instance is infeasible. Let us give a formal outline of the algorithm described above.

Initialization

Put $\mathcal{C}_0 := \emptyset$, $n_0 := 0$, $\mathcal{C}_{y_1} := \{y_1\}$, $n_{y_1} := 1$.

Let $\mathcal{S}_1 := \{\mathbf{0}, y_1\}$.

The main loop:

For all $k = 2, \dots, N$ **do**

$\mathcal{S}_k := \mathcal{S}_{k-1}$.

For all $z \in \mathcal{S}_{k-1}$ **do**

 If \mathcal{S}_k contains z' such that $z' = z + y_k$ **then**

If $n_{z'} < n_z + 1$ **then**

$n_{z'} = n_z + 1$.

$\mathcal{C}_{z'} = \mathcal{C}_z \cup \{y_k\}$.

End if.

Else

$\mathcal{S}_k := \mathcal{S}_k \cup \{z + y_k\}$.

$n_{z+y_k} := n_z + 1$.

$\mathcal{C}_{z+y_k} := \mathcal{C}_z \cup \{y_k\}$.

End if.

End for.

End for.

Search for $z^* \in \mathcal{S}_N$ such that $\frac{\|z^*\|^2}{n_{z^*}} \leq \frac{\alpha}{|\mathcal{Y}|} \|\sum_{y \in \mathcal{Y}} y\|^2$ and n_{z^*} is maximum.

Output z^* if it exists, otherwise report the problem is infeasible.

Taking into account that computing \mathcal{S}_k takes $\mathcal{O}(q \cdot |\mathcal{S}_{k-1}|)$ operations, we have the following

Theorem 2. *If the coordinates of the input vectors from \mathcal{Y} are integer and each of them is at most b by the absolute value then MCSV problem is solvable in $\mathcal{O}(qN(2bN + 1)^q)$ time.*

In the case of fixed dimension q the running time of the algorithm is $\mathcal{O}(N(bN)^q)$, i. e. the problem is solvable in pseudo-polynomial time in this special case.

4 Computational Experiments

This section contains the results of testing the dynamic programming algorithm (DP) proposed in Sect. 3 and the results of COINBONMIN solver (CBM).

For the experiments, the DP algorithm was implemented in C++ and tested on a computer with Intel Core i7-4700 2.40 GHz processor and amount of RAM 4GB. First of all, two series of instances were generated randomly. To generate these series, we fixed parameter $\alpha = 0.1$, the dimension of the space $q = 5$ and number of vectors $N = 1000$. In Series 1, the values of the vector coordinates varied from -1 to 1 , in Series 2 they varied from -5 to 5 and were integers. In both series the coordinates of vectors were generated with uniform distribution.

All testing instances were solved by DP algorithm and by the package COIN-BONMIN, included in the GAMS package, using the quadratic programming model from Sect. 1 (see formulas (2) to (4)).

Table 1. CPU time comparison of the solver CBM and DP algorithm on Series 1

Problem	CBM value	DP value	CBM time	DP time	Problem	CBM value	DP value	CBM time	DP time
1	977	977	106,8	40,4	16	975	975	91,1	32,1
2	971	971	131,5	17,0	17	984	984	132,3	36,3
3	972	972	129,3	65,5	18	986	986	20,7	18,6
4	986	986	17,7	15,2	19	971	971	180,6	55,1
5	986	986	18,7	17,2	20	983	983	97,5	63,1
6	981	981	91,7	23,5	21	978	978	36,7	27,4
7	984	984	55,5	19,7	22	984	984	191,5	39,1
8	965	965	232,3	43,2	23	977	977	64,6	42,5
9	979	979	98,6	57,8	24	958	958	57,6	31,6
10	968	968	99,5	20,7	25	986	986	20,7	18,6
11	970	970	127,7	29,1	26	966	966	77,3	40,4
12	990	990	27,6	21,5	27	965	965	324,9	45,3
13	974	974	25,6	23,1	28	986	986	24,9	22,9
14	964	964	255,1	37,5	29	965	965	65,4	47,4
15	981	981	542,1	46,4	30	973	973	408,7	53,8

The results of the computational experiment for the Series 1 are presented in Table 1. Here and below, we use the bold font to emphasize the best CPU time for each of the instances. For all problems of the series, both algorithms have found optimal solutions. However in all cases, the DP algorithm found the optimal solution faster than CBN. On Series 2, in the majority of the cases DP works faster as well (The results are presented in Table 2). The Wilcoxon signed-rank test showed that the CPU times of the DP and CBN on both Series 1 and Series 2 differ with a significance level less than 5%.

We also made an experiment, with Series 3, based on the historical data on electricity prices from PJM Interconnection (USA), available at <http://www.pjm.com>. The dimension of the space turned out to be exceedingly large for the

Table 2. CPU time comparison of the solver CBM and DP algorithm on Series 2

Problem	CBM value	DP value	CBM value	DP time	Problem	CBM value	DP value	CBM time	DP time
1	990	990	19,1	117,3	16	977	977	23,5	136,9
2	977	977	110,3	87,6	17	990	990	207,7	99,2
3	985	985	224,3	93,1	18	983	983	289,4	78,2
4	963	963	199,6	135,4	19	983	983	17,8	84,3
5	983	983	15,1	105,8	20	968	968	272,3	95,1
6	982	982	62,1	96,8	21	982	982	17,9	137,1
7	975	975	527,6	89,2	22	961	961	635,6	125,0
8	967	967	518,3	137,5	23	984	984	164,6	106,6
9	979	979	124,4	94,8	24	971	971	167,4	98,6
10	978	978	112,5	101,4	25	983	983	28	84,3
11	965	965	65,4	126,5	26	989	989	203,1	124,2
12	967	967	127,9	85,7	27	971	971	140,6	92,3
13	981	981	16,8	87,6	28	987	987	536,6	116,7
14	974	974	178,3	81	29	965	965	25,4	83,5
15	983	983	494,2	96	30	986	986	66,7	64,8

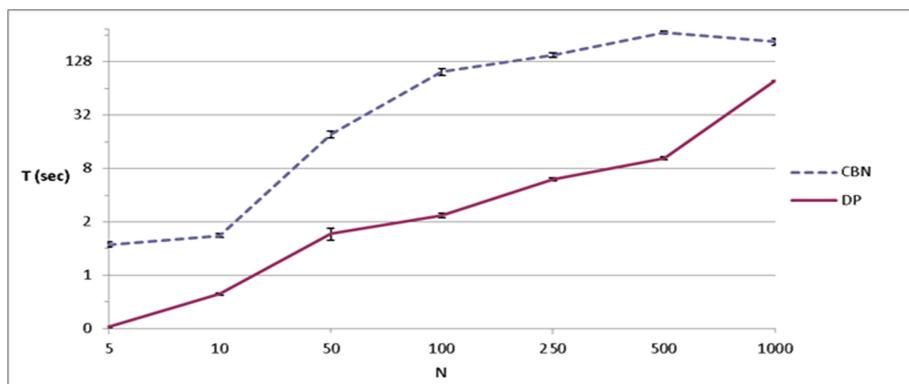
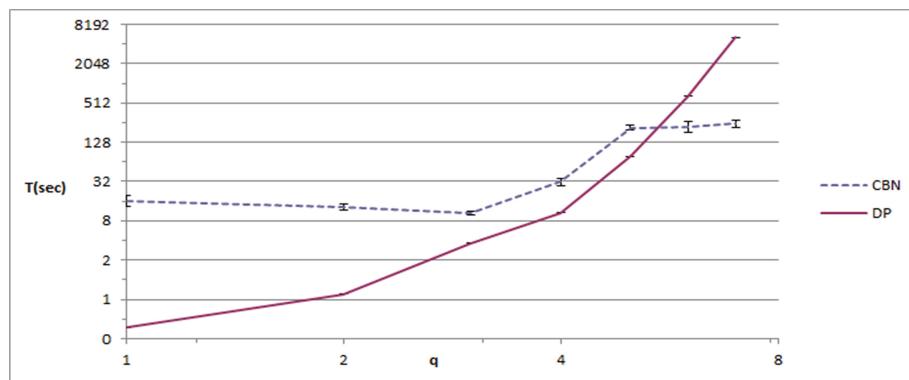
DP algorithm to meet these challenges, while CBM algorithm was able to solve these problems. This is due to a dimension of the space $q = 24$. The value of the α parameter was taken to be 0.1. The results of the experiment with Series 3 are shown in Table 3. The optimal solutions are marked with “*” symbol. It is worth noting that CBM solver could not find the optimal solution to the 4-th instance and managed to find only an approximate solution.

In additional experiments, we generated three series of instances in order to investigate how the values of N, q and α affect the execution time of the DP algorithm and COINBONMIN package. In Series 4, we fixed $q = 5$ and $\alpha = 0.1$ and varied N from 5 to 1000, see Fig. 1. For Series 5 we put $N = 1000$, $\alpha = 0.1$ and varied q from 1 to 7, see Fig. 2. In Series 6, we fixed $q = 5$ and $N = 1000$ and varied parameter α from 0.1 to 0.9, see Fig. 3. Six problem instances were randomly generated and solved for each set of the parameters mentioned above. Average CPU times of both algorithms are presented in Figs. 1, 2 and 3 where the error intervals show the standard error of the mean.

The results of experiments with Series 1–5 indicate that in the cases where the dimensionality of the space and the maximum value of the coordinates of input vectors are not large, the DP algorithm is the most appropriate. However, when the dimension of space increases, it is preferable to use COINBONMIN as a mixed integer quadratically constrained program solver (miqcp mode). Experiments with Series 6 show that the execution time of COINBONMIN solver is not stable w.r.t. variation of α , while the CPU time of the DP algorithm does not depend on this parameter (which clearly agrees with the DP algorithm description).

Table 3. CPU time of the solver CBM for electricity prices “PJM Interconnection”

Problem	CBM value	CBM time	N
1	40*	0.311	43
2	118*	2.293	152
3	177*	2.503	199
4	186	87,984	199
5	223*	0.867	233
6	397*	2,02	408
7	630*	3.686	642
8	625*	2.776	642

**Fig. 1.** The average execution time of the DP algorithm and COINBONMIN package as a function of N **Fig. 2.** The average execution time of the DP algorithm and COINBONMIN package as a function of q

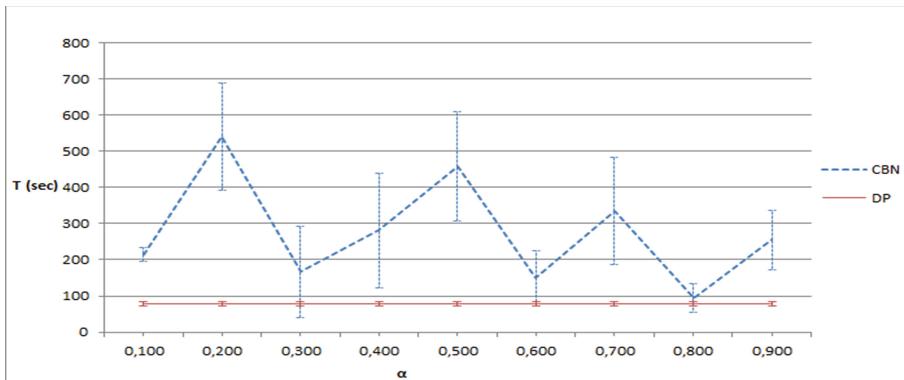


Fig. 3. The average execution time of the DP algorithm and COINBONMIN package as a function of α

5 Conclusions

The problem of finding a maximum cardinality subset of vectors, given a constraint on the normalized squared length of vectors sum is considered for the first time. It is shown that even finding a feasible solution to this problem is NP-hard and an exact dynamic programming algorithm for solving this problem is proposed. We prove a pseudo-polynomial time complexity bound for this algorithm in the special case, where the dimension of the space is bounded from above by a constant and the input data are integer. An alternative approach to solving the problem is based on the mixed integer quadratic programming. Both approaches are compared in a computational experiment. The results of the experiment indicate that in the cases where the dimensionality of the space and the maximum value of the coordinates of input vectors are not large, the dynamic programming algorithm is the most appropriate. However, when the dimension of the space increases, it is preferable to use a mixed integer quadratically constrained program solver, like COINBONMIN.

Acknowledgements. This research is supported by RFBR, projects 15-01-00462, 16-01-00740 and 15-01-00976.

References

1. Ageev, A.A., Kel'manov, A.V., Pyatkin, A.V., Khamidullin, S.A., Shenmaier, V.V.: Polynomial approximation algorithm for the data editing and data cleaning problem. *Pattern Recogn. Image Anal.* **27**(3), 365–370 (2017)
2. Borisovsky, P.A., Eremeev, A.V., Grinkovich, E.B., Klokov, S.A., Vinnikov, A.V.: Trading hubs construction for electricity markets. In: Kallrath, J., Pardalos, P.M., Rebennack, S., Scheidt, M. (eds.) *Optimization in the Energy Industry*, pp. 29–58. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88965-6_3

3. Borisovsky, P.A., Eremeev, A.V., Grinkevich, E.B., Klokov, S.A., Kosarev, N.A.: Trading hubs construction in electricity markets using evolutionary algorithms. *Pattern Recogn. Image Anal.* **24**(2), 270–282 (2014)
4. Eremeev, A.V., Kel'manov, A.V., Pyatkin, A.V.: On complexity of searching a subset of vectors with shortest average under a cardinality restriction. In: Ignatov, D.I., Khachay, M.Y., Labunets, V.G., Loukachevitch, N., Nikolenko, S.I., Panchenko, A., Savchenko, A.V., Vorontsov, K. (eds.) AIST 2016. CCIS, vol. 661, pp. 51–57. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-52920-2_5
5. Eremeev, A.V., Kel'manov, A.V., Pyatkin, A.V.: On the complexity of some Euclidean optimal summing problems. *Dokl. Math.* **93**(3), 286–288 (2016)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, San Francisco (1979)
7. Greco, L.: Robust Methods for Data Reduction. Chapman and Hall/CRC, Boca Raton (2015)
8. NEPOOL Energy Market Hub White Paper and Proposal. Hub Analysis Working Group NEPOOL Markets Committee (2003)
9. Osborne, J.W.: Best Practices in Data Cleaning: A Complete Guide to Everything You Need to Do Before and After Collecting Your Data, 1st edn. SAGE Publication, Inc., Los Angeles (2013)
10. de Waal, T., Pannekoek, J., Scholtus, S.: Handbook of Statistical Data Editing and Imputation. John Wiley and Sons, Inc., Hoboken (2011)

Using Cluster Analysis for Characteristics Detection in Software Defect Reports

Anna Gromova^(✉)

Exactpro, Moscow, Russia

anna.gromova@exactprosystems.com

<https://www.exactprosystems.com/>

Abstract. Computing and predicting testing metrics are highly important software defect management tasks. Before testing metrics are used, one needs to understand and evaluate the full array of software defects found during testing. There can be thousands of software defects found during testing, and it is difficult to process all of them en masse. Cluster analysis solves this problem, as it can compress the data by grouping a set of objects into a cluster. Moreover, clustering helps understand the nature of defects, point out weak functional areas of the software, and improve the testing strategy. This paper introduces a technique used for software defect reports clustering.

Keywords: Software · Defect report · Cluster analysis

1 Introduction

According to a review of research in software defect reporting [25], developing appropriate testing metrics is one of the key open research problems. The research in this area allows predicting how long it will take to fix a bug report, whether it will get rejected, reopened, etc. [1, 11, 26, 28]. Predictions of testing metrics should give project managers a better picture of the risks associated with software defects. As most of the test metrics are business- and task-specific, we need to understand the peculiar properties of defect reports before using any testing metrics.

This task is usually accompanied by several obstacles. Firstly, a project can have thousands of defects, which is hard to process manually. Secondly, we need to evaluate not only the statistical data, but also the heuristic links between the different defect attributes. In this paper, these links are referred to as characteristics of defect reports.

We propose cluster analysis of defect reports as a way to resolve these problems and achieve the prediction task. Clustering compresses data by grouping a set of objects into clusters. This method gives an opportunity to understand the nature of defects. Moreover, analyzing each cluster separately helps with tackling software weaknesses and improving the testing strategy.

Our experimental data consists of 8,583 defect reports that were derived from our projects. Each defect was represented as a set of attributes. The attributes are of numeric, Boolean and categorical types. All the data that we got from the *Component/s* and the *Summary* text fields were transformed into the Boolean type.

In this work, we use the k-means algorithm, setting the number of clusters based on calculating the Silhouette and the Davies-Bouldin indices.

We claim the following contribution in this work:

- Expanding the scope of defect report attributes conventionally taken into account by the researchers in the field and including implicit data to gain a wider perspective in the process of bug examination.
- Calculating the Silhouette and the Davies-Bouldin indices to find the proper number of clusters for the k-means clustering algorithm.
- Providing a description and interpretation of the received clusters.

The remainder of this paper is organized as follows. In Sect. 2, we present the overview of the related work, in Sect. 3, we describe what defect attributes and types of attributes were used in our work and why, and in Sect. 4, we outline the process of clustering. Then, in Sect. 5, we present the results of the experimental evaluations of this technique, and, in Sect. 6, we present the conclusions.

2 Related Work

There are many researchers who try to use cluster analysis in defect management for different objectives.

P.N. Minh proposed to use cluster analysis in order to find defect report duplicates [18]. He extracted defects from three open source projects (Argo UML, Apache and Subversion). He used the comments for mapping, and the *Title* and the *Description* fields for analysis.

Rus et al. [24] used clustering in order to understand the nature of defect reports. They extracted the information about the bugs from Mozilla and applied three algorithms: k-means, normalized cut and size regularized cut. The *Summary* and the *Description* text fields were used in this research. The experiments showed that normalized cut achieved the best results.

Fry and Weimer [9] proposed to use cluster analysis for grouping defect reports. In this research, defects are produced by special tools called defect detectors. These defect detectors generate defects automatically using the program code. However, these tools generate too many defects, making it difficult to process all of them en masse. Cluster analysis can help to triage and fix the defects via aggregation of bug reports.

Nagwani and Bhansali [20] proposed to predict the “complexity” of bugs as it may significantly help to plan the testing workload automatically. For this goal, they calculated the duration (fix time) of bugs extracted from the MySQL repository. Then they clustered defects according to this calculated duration, using the k-means algorithm.

Limsettho et al. [16] proposed to categorize defect reports without a training set. They extracted defect reports from three projects (Lucene, Jackrabbit, HTTP Client) and used cluster analysis to group the bug reports. This clustering was based on the textual similarity of such bug properties as title, description and comments.

Generally, cluster analysis compresses defects by grouping them into sets of similar objects. We can find similar bug reports and then analyze the observed groups of bugs. In all the papers mentioned above, this method proved its effectiveness in finding bug duplicates, automating testing, predicting the testing workload and improving the defect management practices.

However, the majority of them retrieve the text data from the bug reports and ignore other important Boolean, numeric and categorical bug attributes. In this paper, we propose a method of bug report clustering using a set of different bug properties and setting the number of clusters for k-means by calculating the Silhouette and the Davies-Bouldin indices.

3 Background

Every defect can be represented as a set of attributes. The defects submitted to a bug tracking system (BTS) can be downloaded as CSV files for further investigation. We used the defects from our company's projects in a BTS as material for our research.

Among the various kinds of attributes required for cluster analysis of bugs, certain kinds are defined implicitly, such as *the Time to resolve* or *the Area of testing*. QA Leads or Managers create test strategies and test plans, dividing the whole scope of work into several areas of testing connected with each other. As a rule, they do not assign distinct components like separate buttons or functions to QA Engineers, as development team leads usually do. Instead, QA Leads assign the areas of testing or their intersections. For instance, the area of interfaces that includes different gateways, or the area of deployment that includes various scripts, databases, etc. In creating a testing strategy, we do not use separate scripts or gateways as a starting point. Rather, we employ the terms belonging to a higher level of abstraction. An expert should be responsible for defining the number of areas of testing for the project. We used *the Summary* and *the Component/s* text fields from the defect report for the area of testing computation, because these fields include the necessary information about the software component group. We did not use such text fields as Description or Comments because they are aimed at explaining the problem (steps to reproduce, expected and observed behaviour, etc.), which, for our research, is excessive information.

It is worth mentioning that manual classification was only used as the first stage of our research. It is due to manual classification that we are able to use these manually marked data for classifier training. So, classifier learning (with the help of machine learning techniques) is the second stage of the process. The third and final stage, as well as the ultimate goal of our research, is using these trained classifiers for the analysis of new bug reports. Using them allows us

to determine what cluster a bug belongs to and understand its characteristics automatically, without any manual help. Thus, being able to perform automatic classification of defects according to the area of testing will be of much value in the future. For example, we will be able to predict *the Time to Resolve*.

The defect dataset is presented as follows: $D = \{d_1, d_2, \dots, d_j, \dots, d_n\}$, where d_j is a defect, n is the number of defects in the project.

Each defect is described by the following attributes: $d_j = \{\text{Priority}, \text{Status}, \text{Resolution}, \text{Time To Resolve}, \text{Count Of Attachments}, \text{Count Of Comments}, \text{Area}_1, \dots, \text{Area}_k\}$, where k is the number of defined areas of testing.

Every attribute of bug report has defined values in BTS. For example *the status* can have the following values: open, reopened, resolved, closed, etc. The attribute being computed by us, called *The Area of Testing*, also has its own values. They include the area of interfaces, the area of deployment, etc. But for the purposes of this paper, we obfuscated the real values for all attributes by giving them consecutively numbered abstract values instead. The matter is that a bug report is part of software development data and may contain confidential information, so it is considered a trade secret.

The full list of attributes with their descriptions is presented in Table 1.

Table 1. Attribute description

Attribute	Data type	Values	Computing	Comments
Priority	Categorical	priority1, priority2, priority3, priority4		An absolute classification [15]
Status	Categorical	status1, status2, status3, status4, status5, status6		
Resolution	Categorical	resolution1, resolution2, resolution3, resolution4, resolution5, resolution6, resolution7, resolution8, resolution9, resolution10		
Time to resolve	Numeric		Count of days between <i>data created</i> and <i>data resolved</i> . Data should be normalized	The indicator of how expensive a bug report is [14]
Count of attachments	Numeric		Data should be normalized	The attached files improve the quality of defect description [14]
Count of comments	Numeric		Data should be normalized	A large number of comments is an indicator of insufficient defect description [14]
Area i	Boolean	0,1	Classification of defect reports according to the area of testing. If a defect belongs to this area, then the attribute is equal to 1	The area of testing is a group of software components [10]

4 Approach

4.1 Objects

We extracted 8,583 bug reports from our projects in a BTS. We analyzed the defects from two projects. The first project contains 2,795 defects. The distribution of defect reports by the *Priority*, *Status*, *Resolution* and *Area of testing* attributes is illustrated in Fig. 1. Eight areas of testing were defined for the first project. The second project consists of 5,788 bug reports. The distribution of defect reports for this project is presented in Fig. 2. Ten areas of testing were defined for the second project.

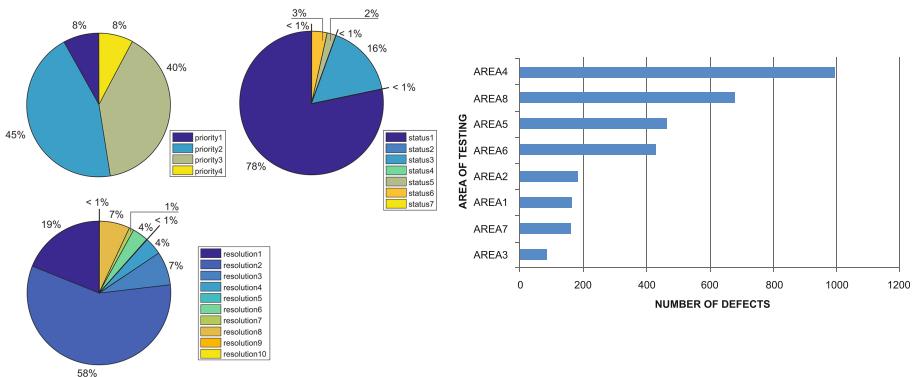


Fig. 1. Distribution of defect reports by the *Priority*, *Status*, *Resolution*, *Area of testing* attributes for Project 1

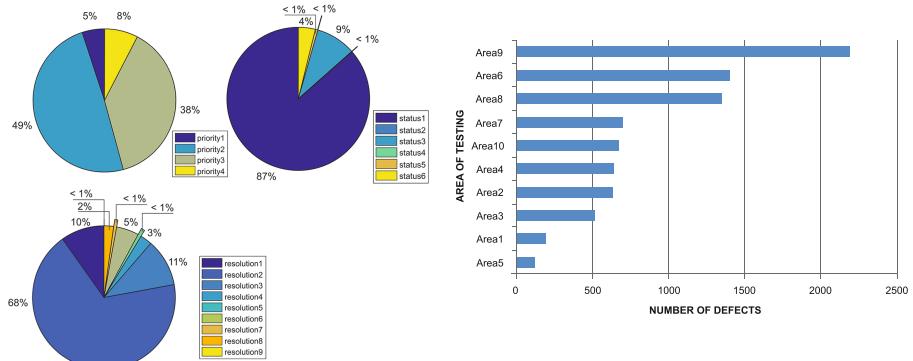


Fig. 2. Distribution of defect reports by the *Priority*, *Status*, *Resolution*, *Area of testing* attributes for Project 2

4.2 Preprocessing

Text fields

For the purposes of our research, we needed the text information from the bug reports. We used the information from *the Summary* and *the Component/s* text fields in order to classify the defects according to the area of testing [10]. Our experts defined the areas of testing for two projects, after which all the defects were manually classified according to the defined areas. For the first project, the expert defined eight areas of testing and for the second project - ten.

Every bug belongs to one or several areas, so this is a classification with overlapping classes. In such a case, we needed to build eight binary classifiers for the first project and ten binary classifiers for the second project. The marked data were used for preprocessing the defect reports. Preprocessing consisted of such steps as tokenization, removal of stop words, and stemming. Every defect is considered a separate document, and a set of defects is considered a corpus. Then we built a Bag-of-words vector space model via TF-IDF indexing [17]. We matched every indexed bug report with its class 0, 1 for each area, where “1” means that defect belongs to a specific *area of testing*. This column was used by the machine learning techniques during the training phase.

In our research, we used six classification methods: logistic regression [8], support vector machines [7], decision tree [22], random forest [2], Bayes net [23] and Naive Bayes [23]. For feature selection, we used such methods as information gain [21], the consistency-based [5] and correlation-based methods [12], and the simplified silhouette filter [4]. It is important to mention that we had classifiers learn both with and without feature selection in order to prove its usefulness.

After that, we compared the F-measure values among all combinations. We also evaluated the performance of the classifiers in the cases of hold-out (we divided the set into two parts: 70% training, 30% testing) and cross-validation (the 10-fold variant). We discovered that the cases without feature selection have lower F-measure values than others. Some of these values are lower than 0.5 (e.g. SVM without feature selection). Thus, feature selection is an integral part of a successful classification process. In addition, we found out that the following combinations of the classifiers and the feature selection methods have the best results for both projects: random forest and the consistency-based method, support vector machines and the consistency-based method. Another significant conclusion we have made is that cross-validation allows improving the results. The F-measure values received are presented in [10]. These data include the best combinations of methods in the case of cross-validation. Finally, we found out that, in the cases with feature selection, the lowest F-measure values are typical for the Bayes classifier.

Numeric fields

The numeric attributes were normalized via zero-mean normalization (standardization). This method standardizes the features so that they are centered around 0 with a standard deviation of 1 [27]. The samples are calculated as follows:

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

where μ is the mean and σ is the standard deviation from the mean.

We also determined whether the variables are highly correlating or not [19]. The Pearson correlation coefficient between the clustering variables is not higher than 0.5, therefore, according to the Chaddock scale, this correlation is loose or even very loose [3].

4.3 Clustering

The task of clustering is to construct set $C = \{c_1, c_2, \dots, c_k, \dots, c_g\}$ where c_k is the cluster that contains similar objects from dataset D. c_k is defined as follows: $c_k = \{d_j, d_q \in D, \text{distance}(d_j, d_q) < \sigma\}$, where σ is a value that defines the proximity of objects to be included in one cluster. We used the k-means algorithm for clustering [13, 27]. Algorithm k-means is fast and simple. Also K-means models clusters using the simplest model - a centroid. So massive data are reduced to centroids and become easier for interpretation.

We used the Silhouette and the Davies-Bouldin indices [6] in order to define the correct count of clusters. According to the received results, the optimal count for clustering of this data is six for Project 1, and five for Project 2. The results of the validity indices are presented in Table 2.

Table 2. The results of the validity indices

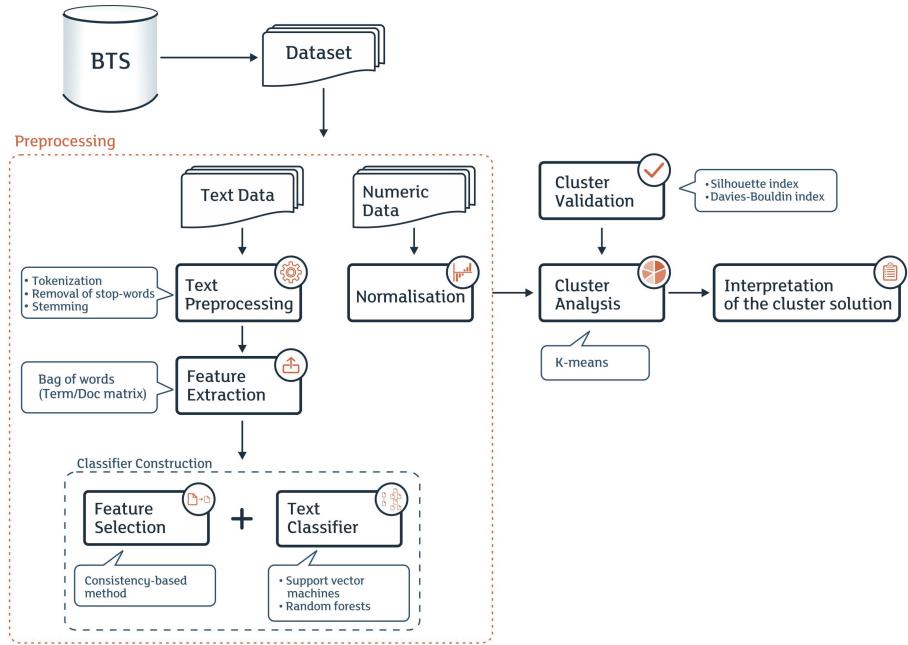
Index	Count of clusters/ Project	2	3	4	5	6	7	8	9
Silhouette Index	1	0.9993	0.9996	0.9999	1	1	0.6489	0.6335	0.6375
Davies-Bouldin Index	1	0.2733	0.2445	0.1098	0.0488	4.5635e-04	0.2367	0.3278	0.5492
Silhouette Index	2	0.9997	0.9997	0.9999	1	0.57	0.6284	0.6381	0.6002
Davies-Bouldin Index	2	0.2364	0.3939	0.1413	2.9406e-04	0.3364	0.4304	0.5329	0.6636

The approach is illustrated in Fig. 3.

5 Results

The final centroids are presented in Tables 3 and 4. According to zero-mean normalization, any value that is lower than 0 is lower than the mean value; any value that is higher than 0 is higher than the mean value. For the rows with areas of testing, “0” means that bugs of this cluster does not belong to the mentioned area of testing, “1” means that bug belongs.

Based on the final centroids of the clusters, let's look into the characteristics of the defects.

**Fig. 3.** Clustering of defect reports**Table 3.** Final centroids of the first project

	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Number of bugs/attribute	653	175	909	426	208	424
Priority	Priority3	Priority3	Priority3	Priority2	Priority2	Priority2
Status	Status1	Status1	Status1	Status1	Status1	Status1
Resolution	Resolution2	Resolution2	Resolution2	Resolution2	Resolution2	Resolution2
Time to resolve	-0.112	0.2846	0.075	-0.2365	-0.2172	0.2385
Count of comments	-0.0398	-0.3405	-0.0426	0.1529	-0.2418	0.2581
Count of attachments	-0.1479	-0.1257	0.1182	0.1012	-0.1763	0.0111
Area 1	0	0	0	0	0	0
Area 2	0	0	0	0	1	0
Area 3	0	0	0	0	0	0
Area 4	0	0	1	0	0	0
Area 5	0	0	0	0	0	1
Area 6	0	0	0	1	0	0
Area 7	0	1	0	0	0	0
Area 8	1	0	0	0	0	0

According to Table 3, the “fastest-to-resolve” defects belong to **Clusters 3, 4, and 0**. **Cluster 3** contains the bugs with a large number of collateral comments. This may be explained by the bug being poorly documented by the QA, or by the fact that the developers fixed the symptoms of the bug instead of eliminating the root cause. This, in turn, may result from *area 6* being a complex area. **Clusters 0 and 4** contain the bugs that do not require collateral comments and are easy to resolve. This may stem from these areas, i.e. *area 8* and *area 2*, being simpler. The difference between the bugs of these two clusters is in the levels of priority.

According to Table 3, the “longest-to-resolve” defects belong to **Clusters 1 and 5**. **Cluster 5** comprises the defects that are hard to describe and/or resolve. Since this cluster is related to *area 5*, this might be a signal of the complexity of this area. **Cluster 1** contains the “longest-to-resolve” defects, but the lowest number of collateral comments among all the clusters. As these defects correspond to *area 7*, this area might contain easy-to-resolve, but low-priority bugs, that is also confirmed by the difference in the *priority* number between **Clusters 5 and 0**.

The *time to resolve* in **Cluster 2** is close to the mean. The number of collateral comments in this area is low, but the number of attached files is high. Since this cluster is related to *area 4*, this area might require several attachments, such as transaction logs, screenshots, etc.

Table 4. Final centroids of the second project

	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Number of bugs/attribute	578	1855	2051	659	645
Priority	Priority3	Priority2	Priority2	Priority3	Priority3
Status	Status3	Status1	Status1	Status1	Status1
Resolution	Resolution1	Resolution2	Resolution2	Resolution2	Resolution2
Time to resolve	-0.4452	-0.0537	-0.2282	0.8606	0.3999
Count of comments	0.5361	-0.1157	-0.1576	-0.1688	0.526
Count of attachments	0.0263	0.1243	-0.181	0.0025	0.1921
Area 1	0	0	0	0	0
Area 2	0	0	0	0	0
Area 3	0	0	0	0	0
Area 4	0	0	0	0	0
Area 5	0	0	0	0	0
Area 6	0	0	0	0	1
Area 7	0	0	0	0	0
Area 8	0	0	0	1	0
Area 9	0	1	0	0	0
Area 10	0	0	0	0	0

According to Table 4, the “fastest-to-resolve” defects belong to **Clusters 0, 1, and 2**. **Cluster 0** contains the bugs with a large number of collateral comments, but, at the same time, *the Priority* and *Status* of this cluster differ from **Clusters 1** and **2**. Thus, the bugs of this cluster are in a peculiar state, that being a particular phase of a bug’s life cycle. In addition, **Cluster 0** does not have a pronounced *area of testing*. The majority of defects in this cluster belong to *areas 6, 8, and 9*. **Cluster 1** has few comments, but does have attachment files. This cluster corresponds to *area 9*. **Cluster 2** contains the bugs that do not require collateral comments or attachments, so they are easy to resolve. **Cluster 2** does not have a specific **area of testing**. However, most of its defects belong to *areas 2, 4, 6, and 8*.

According to Table 4, the “longest-to-resolve” defects belong to **Clusters 3 and 4**. **Cluster 3** contains the “longest-to-resolve” defects, but the lowest number of collateral comments among all the clusters. As these defects correspond to *area 8*, this area might contain easy-to-resolve, but low-priority bugs. **Cluster 4** comprises the defects that are hard to describe and/or resolve. Since this cluster is related to *area 6*, this might be a signal of the complexity of this area. In some cases, such defects require fixing, but this activity can influence the other areas, so the time to resolve gets extended.

From this we can conclude that the clusters of the first and the second projects constructed during the research are strongly connected to the areas of testing. Some areas require more time to resolve and/or more additional information in the description. Other areas need less time to resolve. So the specificity of the area defines the nature of defects and allows us to correct our testing strategy.

6 Conclusion

This paper is devoted to clustering software defects.

We made use of an extraordinary set of attributes for the cluster analysis that we performed, which distinguishes our paper from other research in the field. In addition to taking into account the numeric and the categorical attributes that are described explicitly, we also used the following implicit attributes: *the Time to resolve* and *the Area of testing*. These implicit attributes, especially the latter, require additional computing, which calls for a preliminary classification of defects. In this work, we used the k-means algorithm, setting the number of clusters by calculating the Silhouette and the Davies-Bouldin indices.

Clustering provides an opportunity to understand the nature of defects and the complexity of different testing areas, as well as to improve the testing strategy.

In the nearest future, we plan to use this method to build an automated recommendation system for Project Managers and QA Team Leads, to improve the existing processes of developing the testing strategies and plans. Also we plan to analyze threats to validity.

References

1. Bhattacharya, P., Neamtiu, I.: Bug-fix time prediction models: Can we do better? In: Proceedings of 8th Working Conference Mining Software Repositories, New York, pp. 207–210. ACM (2011)
2. Breiman, L.: Random forests. *J. Mach. Learn.* **45**(1), 5–32 (2001)
3. Chaddock, R.E.: Principles and Methods of Statistics, 1st edn. Houghton Mifflin Company, The Riverside Press, Cambridge (1952)
4. Coves, T.F., Hruschka, E.R.: Towards improving cluster-based feature selection with a simplified silhouette filter. *Inf. Sci.* **181**(18), 3766–3782 (2011)
5. Dash, M., Liu, H.: Consistency-based search in feature selection. *J. Artif. Intel.* **151**(1–2), 155–176 (2003)
6. Desgraupes, B.: Clustering Indices. Journal of University Paris Ouest - Lab ModalX, pp. 1–34 (2013)
7. Durgesh, K.S., Lekha, B.: Data classification using support vector machine. *J. Theor. Appl. Inf. Technol.* **12**(1), 1–7 (2010)
8. Freund, R.J., Wilson, W.J.: Regression Analysis: Statistical Modeling of a Response Variable. Academic Press, San Diego (1998)
9. Fry, Z.P., Weimer, W.: Clustering static analysis defect reports to reduce maintenance costs. In: Proceedings of Working Conference on Reverse Engineering, WCRE, pp. 282–291 (2013)
10. Gromova, A.: Defect Report Classification in Accordance with Areas of Testing/Proceedings of TMPA 2017 Conference. To be published in Springer CCIS series in 2017
11. Guo, P.J., Zimmermann, T., Nagappan, N., Murphy, B.: Characterizing and predicting which bugs get fixed: an empirical study of microsoft windows. In: Proceedings of 32nd ACM/IEEE International Conference Software Engineering, vol. 1, series ICSE 2010, New York, pp. 495–504. ACM (2010)
12. Hall, M.A.: Correlation-based Feature Subset Selection for Machine Learning. Hamilton, New Zealand (1998)
13. Hartigan, J., Wong, M.: A k-means clustering algorithm. *Appl. Stat.* **28**(1), 100–108 (1979)
14. Hooimeijer, P., Weimer, W.: Modeling bug report quality. In: ASE 2007: Proceedings of the Twenty-second IEEE/ACM International Conference on Automated Software Engineering, pp. 34–43 (2007)
15. Lamkanfi, A., Demeyer, S., Soetens, Q., Verdonck, T.: Comparing mining algorithms for predicting the severity of a reported bug. In: Proceedings of 15th European Conference Software Maintenance Reengineering (CSMR), pp. 249–258 (2011)
16. Limsettho, N., Hata, H., Monden, A., Matsumoto, K.: Automatic unsupervised bug report categorization. In: 2014 6th International Workshop on Empirical Software Engineering in Practice, pp. 7–12 (2014)
17. Manning, C.D., Raghavan, P., Schutze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)
18. Minh, P.N.: An approach to detecting duplicate bug reports using n-gram features and cluster chrinkage technique. *Int. J. Sci. Res. Publ. (IJSRP)* **4**(5), 89–100 (2014)
19. Sarstedt, M., Mooi, E.: A Concise Guide To Market Research. The Process, Data, and Methods Using IBM SPSS Statistics. Springer, Heidelberg (2011). <https://doi.org/10.1007/978-3-642-53965-7>

20. Nagwani, N.K., Bhansali, A.: A data mining model to predict software bug complexity using bug estimation and clustering. In: Proceedings of 2010 International Conference Recent Trends Information Telecommunication Computer series ITC 2010, pp. 13–17, IEEE Computer Society, Washington, DC. (2010)
21. Niccolosi, N.: Feature Selection Methods for Text Classification (2008)
22. Quinlan, J.R.: C4.5: Programs For Machine Learning. Morgan Kaufman, San Francisco (1993)
23. Rish, I.: An empirical study of the naive bayes classifier. In: Proceedings of the IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, pp. 41–46 (2001)
24. Rus, V., Nan, X., Shiva, S., Chen, Y.: Clustering of defect reports using graph partitioning algorithms. In: Proceedings of the 21st International Conference on Software Engineering and Knowledge Engineering, pp. 442–445 (2009)
25. Strate, J.D., Laplante, P.A.: A literature review of research in software defect reporting. *IEEE Trans. Reliab.* **62**, 444–454 (2013)
26. Weiss, C., Premraj, R., Zimmermann, T., Zeller, A.: How long will it take to fix this bug? In: Proceedings 4th International Workshop Mining Software Repositories, series. MSR 2007, vol. 1, IEEE Computer Society, Washington, DC. (2007)
27. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufman, San Francisco (2005)
28. Zhou, Y., Tong, Y., Gu, R., Gall, H.C.: Combining text mining and data mining for bug report classification. In: Proceedings of 30th International Conference on Software Maintenance and Evolution (ICSM/ICSME), pp. 311–320. IEEE (2014)

A Machine Learning Approach to Enhanced Oil Recovery Prediction

Fedor Krasnov^(✉) , Nikolay Glavnov, and Alexander Sitnikov

Gazpromneft NTC, 75-79 Moika River emb., St. Petersburg 190000, Russia

krasnov.fv@gazprom-neft.ru

<http://ntc.gazprom-neft.ru>

Abstract. In a number of computational experiments, a meta-algorithm is used to solve the problems of the oil and gas industry. Such experiments begin in the hydrodynamic simulator, where the value of the function is calculated for specific nodal values of the parameters based on the physical laws of fluid flow through porous media. Then, the values of the function are calculated, either on a more detailed set of parameter values, or for parameter values that go beyond the nodal values.

Among other purposes, such an approach is used to calculate incremental oil production resulting from the application of various methods of enhanced oil recovery (EOR).

The authors found out that in comparison with the traditional computational experiments on a regular grid, computation using machine learning algorithms could prove more productive.

Keywords: Enhanced oil recovery · EOR · Random forest
Regular grid interpolation

1 Proxy Model Approach

One of the main reasons for the appearance of the meta-algorithms in Oil&Gaz industry is the limitations on the speed of hydrodynamic modeling. In the future, when any specialist of an organization will be able to vary the values of the parameters at any time and within a wide range and get the required values of the function in near-real-time mode, the need for a meta-algorithm will disappear. Meanwhile, it takes experts hours or even days to perform modeling for one set of parameters on high-cost, high-performance clusters (HPC). Thus, there is a need for astute preparation of data for further processing. Since the need to change the parameters can occur several times a day and with a whole variety of specialists, an efficient meta-algorithm is an urgent necessity.

As a result of applying the meta-algorithm, a model - sometimes called a proxy model - is obtained in [1, 2]. At the input, the proxy model receives a set of parameters. Then, it outputs the value of the physical function from these parameters, performing interpolation or extrapolation based on the previously

calculated values of the function in the nodal values of the parameters. The completed proxy model does not require a large amount of computational resources and works in a close-to-real-time mode. It yields immediate results.

In this article, we consider two different approaches to constructing a proxy model, using the example of a computational experiment to study the increment in oil production using miscible displacement by carbon dioxide injection.

2 Hydrodynamic Simulation

To obtain the first estimate of additional oil production from tertiary methods for increasing oil recovery, representative curves in the hydrocarbon pore volume injection and the enhanced oil recovery coordinates are used.

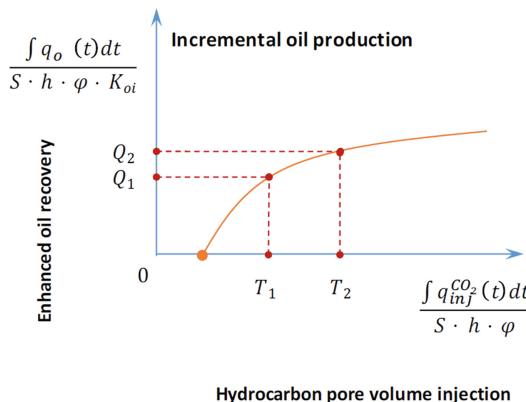


Fig. 1. Representative curve of enhanced oil recovery versus HCPVI.

These curves on Figs. 1 and 2 are most often obtained as a result of statistical analysis of the actually implemented field projects, or from simplified analytical dependencies and, less often, from the results of multivariate calculations on synthetic simulation models. The latter method of obtaining representative curves for the technology of alternating injection of carbon dioxide and water into the reservoir with miscibility was applied in this article.

The simulation was carried out on an Eclipse 300 composite simulator (Schlumberger) which allows to reproduce the process of miscible displacement. The model is a segment of a five-point element of the development system, with vertical wells in the corners. In our experiment, additional oil production is calculated by varying the following parameters:

- Oil properties (density, viscosity, saturation pressure). In order to take into account the influence of the properties of the reservoir system on the displacement efficiency, three models of reservoir oil were created with characteristics covering the entire range (223 objects) of the properties of the reservoir oil of the available oil samples.

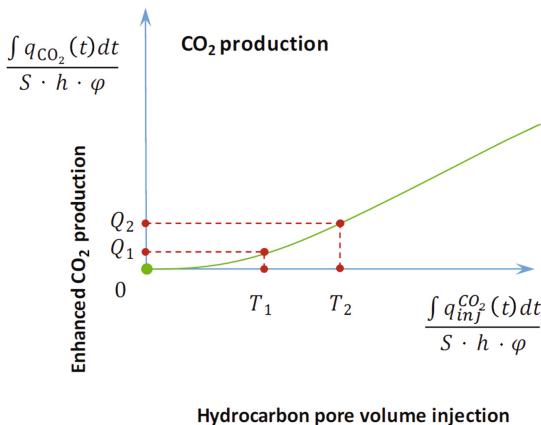


Fig. 2. Representative curve of enhanced CO₂ production versus HCPVI.

- The actual value of the residual oil saturation, obtained from the results of core filtration experiments in the oil-water system, determines the amount of oil that is not attracted by water flooding. During carbon dioxide displacement, a decrease in the residual oil saturation is detected due to a decrease in the tension between the displacing agent and the oil, accompanied by the dissolution process.
- Heterogeneity of permeability: Based on the results of interpretation of logs of exploratory wells through the formula for determining the Dykstra-Parsons coefficient, the values for all the considered objects are calculated. For the sake of variety, four values with almost uniform coverage of the whole interval are picked out.
- Relative phase permeability: to determine the nodal values of the endpoints of the relative phase permeability and the values of the residual water and oil saturation, the results of the laboratory core studies are generalized. To cover the whole range of phase permeability by modeling, the values of the maximum relative phase permeability in gas and water were chosen to correspond to the average value, as well as to two values close to the maximum and the minimum values
- The current oil saturation: in the calculations, the degree of production of stocks was computed by changing the initial oil saturation of the grid when the model was initialized. Three values were identified: the first-production fluid, the average yield, and the produced object.

A total of 324 simulation models were generated. Based on them, 972 calculations were performed (3 per each model). The results of the simulations were grouped into one summary database, in which up to 486 representative curves were processed.

Above on Fig. 3 are the statistics of the time spent on calculating one variant, the average value of which was 90 min.

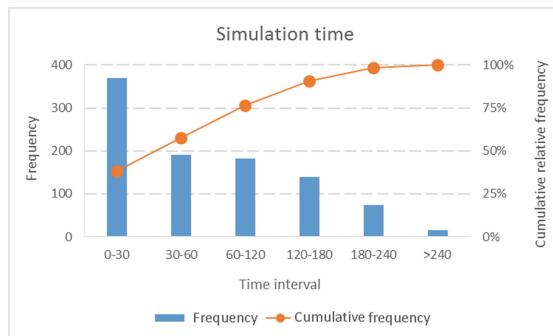


Fig. 3. Simulation time.

2.1 An Approach to Creating a Proxy Model Based on Multidimensional Linear Interpolation

One of the approaches to creating a proxy model is linear multidimensional interpolation described in [3]. To understand it, we will consider the parameter space as a multidimensional cube, in which each dimension is formed by a vector of values of one parameter. Then, the resulting function can also be represented as a vector. The process of creating a proxy-model will contain the following steps:

1. Reading the parameters and values of the function from the results of hydrodynamic modeling;
2. Vectorizing the parameters and the resulting function;
3. Constructing a multidimensional cube of parameters;
4. Constructing an interpolation function;
5. Determining the dimensions of new parameter vectors;
6. Creating the new parameter vectors;
7. Performing interpolation of new parameter vectors by means of an interpolation function;
8. Exporting the received proxy-model in a format convenient for use.

The essence of modeling based on multidimensional linear interpolation is to choose the step of changing new parameters so that the resulting parameter vectors would have the nodal values in their composition and cover the range, necessary for the model, with a sufficient amount of steps. In other words, if you have Parameter P1 with Dimension 3, for which calculations are made in the nodal values of 0.1, 0.5, 0.9, and there is a need for the values of the function at 0.4 and 0.8, then for the new vector, it will be sufficient to select Step 0.1 and Dimension 9. Thus, the parameters are meshed on a regular grid.

In the case when the dimension of the parameter space is greater than 2, we can no longer apply the Spline methods described, among other resources, in [5]. In our case, the dimension of the parameter space is 6 (including the pumped pore volume parameter).

In addition, it is worth noting that the choice of the step should be made, taking into account the capabilities of the computing resources. The vectorized parameter space, represented as a multidimensional array of rational numbers, must correspond to the sizes of the available server RAM for calculations. Under the finished proxy model, in our case, we can understand an MS Excel table with seven columns: six for input parameters and one for the resulting function. Such a presentation is as clear as possible to a wide range of specialists within an organization and allows further research based on the proxy mode data. Figure 4 below shows the dependence of additional production on residual oil saturation and heterogeneity of permeability, with the other parameters having fixed values.

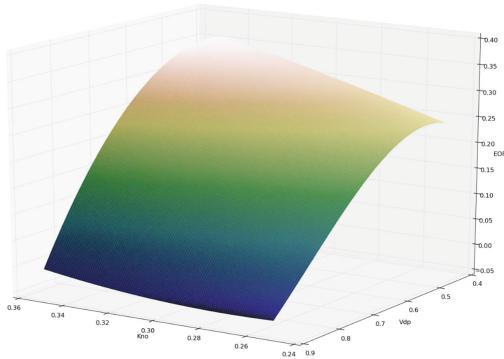


Fig. 4. Additional oil production on a regular interpolation grid model ($Pvt = 1.00$, $KwKg = 0.05$, $Knn = 0.40$, $Hcpv = 145$).

2.2 An Approach to Creating a Proxy Model Based on Machine Learning Methods

Our task belongs to the class of regression-building tasks, from the point of view of machine learning methods. One common and universal regressor is Random Forest - a method coined by Breiman [7]. Random forest is a set of decision trees. In a regression problem, their answers are averaged, in the classification problem - a decision is made by voting on the majority. All trees are constructed independently according to the following scheme [6]:

- A subsample of a training sample of a certain size is chosen. Subsequently, a decision tree is constructed on it (there is a separate subsample for each tree);
- For the construction of each splitting in the tree, a certain number of random features sets is observed. A separate random features set is defined for each new splitting;
- Finally, the best feature is selected based on a predetermined criterion and the splitting goes on according to that criterion. In the original algorithm,

the tree is constructed until the subsample is exhausted, and until the representatives of only one class remain in the leaves. However, in the modern implementations, there are parameters that limit the height of the tree, the number of objects in the leaves, and the number of objects in the subsample, under which the splitting is performed.

This construction scheme corresponds to the main principle of ensemble training [9] - the construction of a machine learning algorithm based on several, in this case, decision trees: the basic algorithms must be good and diverse.

In the above mentioned formulation of the problem of predicting additional oil production, we are training the regressor on the six available parameters and the values of the additional oil recovery factor. Then, we use the resulting regression model for the calculation of values of the additional oil recovery factor based on the new parameter values.

When evaluating the accuracy of the model by predicting the values of the known parameters, the determination coefficient (scaling R-squared) is 0.99 for a test sample of 100 parameter sets. We can also immediately distribute the features by degree of importance (Table 1):

Table 1. Feature importance

Feature	Importance
Oil properties (density, viscosity, saturation pressure)	0.016
The actual value of the residual oil saturation	0.032
Heterogeneity of permeability	0.488
Relative phase permeability	0.041
Current oil saturation	0.026
Pore volume injection (time)	0.397

The accuracy of the result will depend on the Heterogeneity of permeability (Vdp) and Pore volume injection (Time) much more than on the other parameters, as shown in Fig. 5.

You can also evaluate the effect of the number of hydrodynamic simulations on the accuracy of the prediction result.

3 Computational Methods and Algorithms

For calculations, the Python environment was chosen. The choice of Python owes to its extensive capabilities for working with data arrays as matrices, provided by the NumPy library [13]. To work on exporting and importing data to the MS Excel format, the Pandas library [14] was used. For the interpolation of multidimensional surfaces, the SciPy library classes were used. 3D surfaces

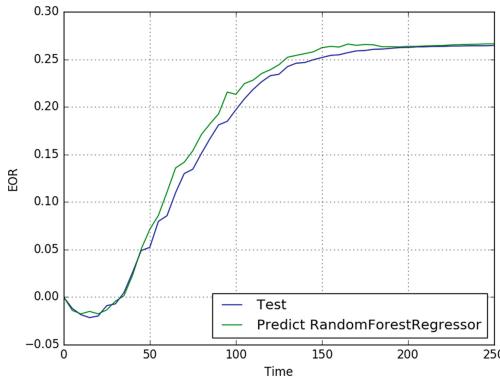


Fig. 5. Additional oil recovery factor on the random forest model.

are rendered using the Matplotlib library. As a software implementation of multidimensional linear interpolation, the Regular Grid Interpolator method from the SciPy library was selected [11,12]. One of the advantages of this method is that it uses the possibilities of a regular grid, instead of the resource-intensive triangulation of the parameter space. Random Forest uses the implementation of scikit-learn [10].

The performance of the libraries used fell within the requirements of the “on demand” computing time. Calculations were performed on a 64-core Linux-run OS (CentOS 7). For the fullest possible use of Multi-core processors, the authors used the Math Kernel Library.

4 Conclusions and Future Directions of Research

It is important to note that this approach should be applied on a regional scale. The authors made the calculation for the entire range of parameter values of the Gazprom Neft fields. In other words, having calculated the additional oil recovery factor once, you can continue working by proxy models (MS Excel tables) without resorting to more calculations, but simply by finding the required set of parameters and the corresponding additional oil recovery factor.

Multidimensional regression using the Random Forest method is a modern and high-performance tool. It meets the requirements of the task of constructing proxy models for calculating the additional oil recovery factor in the range of properties covering all Gazprom Neft fields. It is important to notice complete continuity with respect to multidimensional linear interpolation: on the same data, the same results are obtained with accuracy. Multidimensional regression using the Random Forest method has a number of additional advantages over the method of multidimensional linear interpolation. Namely:

- An ability to take into account the importance of the parameters.
- An ability to determine a sufficient number of calculations of the hydrodynamic models, based on the required accuracy.

The main conclusion of this article is a significant simplification of computations, a significant reduction in the requirements for computational resources and achievement of better predictability of the simulated function, when applying machine learning methods.

References

1. Guo, Z., Reynolds, A.C., Zhao, H.: A Physics-Based Data-Driven Model for History-Matching, Prediction and Characterization of Waterflooding Performance. Society of Petroleum Engineers. <https://doi.org/10.2118/182660-MS>
2. Shehata, A.M., El-banbi, A.H., Sayyoush, H.: Guidelines to Optimize CO₂ EOR in Heterogeneous Reservoirs. Society of Petroleum Engineers. <https://doi.org/10.2118/151871-MS>
3. Weiser, A., Zarantonello, S.E.: A note on piecewise linear and multilinear table interpolation in many dimensions. *Math. Comput.* **50**(181), 189–196 (1988)
4. Ghassemzadeh, S., Charkhi, A.H.: Optimization of integrated production system using advanced proxy based models. *J. Nat. Gas Sci. Eng.* **35**, 89–96 (2016). ISSN 1875-5100
5. Dierckx, P.: Curve and Surface Fitting With Splines Monographs on Numerical Analysis. Oxford University Press, New York (1993)
6. Dyakonov, A.: Blog “Random Forest”, 14 November 2016. <https://alexanderdyakonov.wordpress.com>
7. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
8. Gashler, M., Giraud-Carrier, C., Martinez, T.: Decision tree ensemble: small heterogeneous is better than large homogeneous. In: The Seventh International Conference on Machine Learning and Applications, pp. 900–905 (2008). <https://doi.org/10.1109/ICMLA.2008.154>
9. Opitz, D., Maclin, R.: Popular ensemble methods: an empirical study. *J. Artif. Intell. Res.* **11**, 169–198 (1999). <https://doi.org/10.1613/jair.614>
10. Pedregosa, F., et al.: Scikit-learn machine learning in python. *JMLR* **12**, 2825–2830 (2011)
11. Oliphant, T.E.: Python for scientific computing. *Comput. Sci. Eng.* **9**, 10–20 (2007). <https://doi.org/10.1109/MCSE.2007.58>
12. Jarrod Millman, K., Aivazis, M.: Python for scientists and engineers. *Comput. Sci. Eng.* **13**, 9–12 (2011). <https://doi.org/10.1109/MCSE.2011.36>
13. van der Walt, S., Colbert, S.C., Varoquaux, G.: The NumPy array: a structure for efficient numerical computation. *Comput. Sci. Eng.* **13**, 22–30 (2011). <https://doi.org/10.1109/MCSE.2011.37>
14. McKinney, W.: Data structures for statistical computing in python. In: Proceedings of the 9th Python in Science Conference, pp. 51–56 (2010)

An Approach to Establishing the Correspondence of Spatial Objects on Heterogeneous Maps Based on Methods of Computational Topology

Sergey Eremeev, Kirill Kuptsov^(✉), and Semyon Romanov

Vladimir State University, Vladimir, Russia

sv-eremeev@yandex.ru, kirill-kuptsov@rambler.ru, cwwc@bk.ru

Abstract. The topical problem of automatic establishing the correspondence of spatial objects on different maps of the same terrain without a priori information about key points is considered in the article. The basis of the algorithm is the methods of persistent homology which allow us to identify objects with topological deformations, but with the preservation of the structure of the object. These properties are manifested when displaying objects on maps of different scales or for different periods of time. The results of studies on the implementation of the algorithm for comparing maps from natural objects and for data analysis in municipal geographic information systems are shown.

Keywords: Barcode · Computational topology
Maps of different scales · Spatial objects · Topological relationships

1 Introduction

Comparison of spatial objects on two maps of the same area is an urgent task [1–3]. It has a set of different applications such as updating of maps, comparison of multi-scale maps, object search on a map, selection of similar objects according to certain features [4–6]. Maps can store the heterogeneous information presented in a raster or vectorial form. It is possible to select the following approaches for the decision of this task. Many software applications used correlative approaches to accomplishing this task. They are based on the computation of correlation coefficient between the compared images [7]. Also, there is a number of algorithms and approaches which apply geometrical features for comparing similar objects. Such characteristics as the center of masses, the area and perimeter of a convex shell, and also their relation, etc. are calculated. It is expedient to use these methods when objects are identical to each other and these characteristics remain in the case of affine transformations. Cartographical objects which are located at different scales or change in time can have another form, but it can be similar to the form of a compared object.

After generalization spatial objects become simpler. A source object is deformed. That complicates the use of standard algorithms for comparison of objects at different scales. However, the same object saves the structure and global topological features in the case of generalization. Thus, it is natural to use topological object properties which are invariant to similar deformations and distortions [8, 9].

Also very often key points apply in the case of comparison of heterogeneous maps, for example, a point at the central intersection. Such comparison of maps requires at least two points. However, most often these key points need to be marked manually. Besides, there can be big errors while comparing objects.

Thus, the task of the comparison of spatial objects with small deformations without the use of key points is considered in the article. In order to accomplish this task, it is offered to take methods of persistent homology which consider topological properties of a set of points as a basis [10–12].

2 Methodology

The persistent homology belongs to methods of topological data analysis [13–15]. It begins to be used widely in different areas such as image or signal processing, the analysis of DNA, cluster analysis, text analysis [16, 17]. The essence of the method is in finding regularities from small-size data, i.e. to reveal such structures which will steadily remain in the case of topological deformations and distortions.

Data can be presented in a different way. We consider them as a set of points. Mathematically the method of persistent homology can be described as follows. The radius r circle is built around each point from a data set $V = \{v_{\alpha_1}, v_{\alpha_2}, \dots, v_{\alpha_n}\}$. Vertices which are inside the circle are connected by an edge. If three vertices are inside the circle, then the triangle with the filled internal part is formed. Thus, we receive simplexes $\sigma_n = \langle v_{\alpha_1}, v_{\alpha_2}, \dots, v_{\alpha_n} \rangle$. We will consider the following simplexes: point, line and triangle (Fig. 1(a–c)), i.e. $\sigma_1 = \langle v_{\alpha_1} \rangle$, $\sigma_2 = \langle v_{\alpha_1}, v_{\alpha_2} \rangle$ and $\sigma_3 = \langle v_{\alpha_1}, v_{\alpha_2}, v_{\alpha_3} \rangle$.

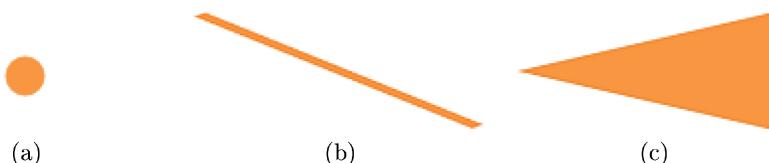


Fig. 1. Simplexes in the form of a point, a line and a triangle with the filled internal part.

The simplicial complex K represents a combination of all simplexes on radius r provided that the boundary of each simplex belongs to K and for two any simplexes this is correct: $\sigma_1 \cap \sigma_2 = \emptyset$ or $\sigma_1 \cap \sigma_2$ have the general edge $\sigma_1, \sigma_2 \in K$.

Besides, topological properties such as quantity of components of connectivity and quantity of holes for each simplicial complex K are calculated. This information is saved in a barcode. Edelsbrunner proposes an algorithm for the computation of a barcode [13].

Initial distance r is a minimum distance between objects on a map and a maximum distance is equal to the greatest distance between objects respectively. The number of simplexes and topological features changes in the case of an increase in distance. The number of simplexes increases, making new components of connectivity and holes afterwards integrating them. The number of components of connectivity is equal to one on the longest distance r as all available components are integrated into one.

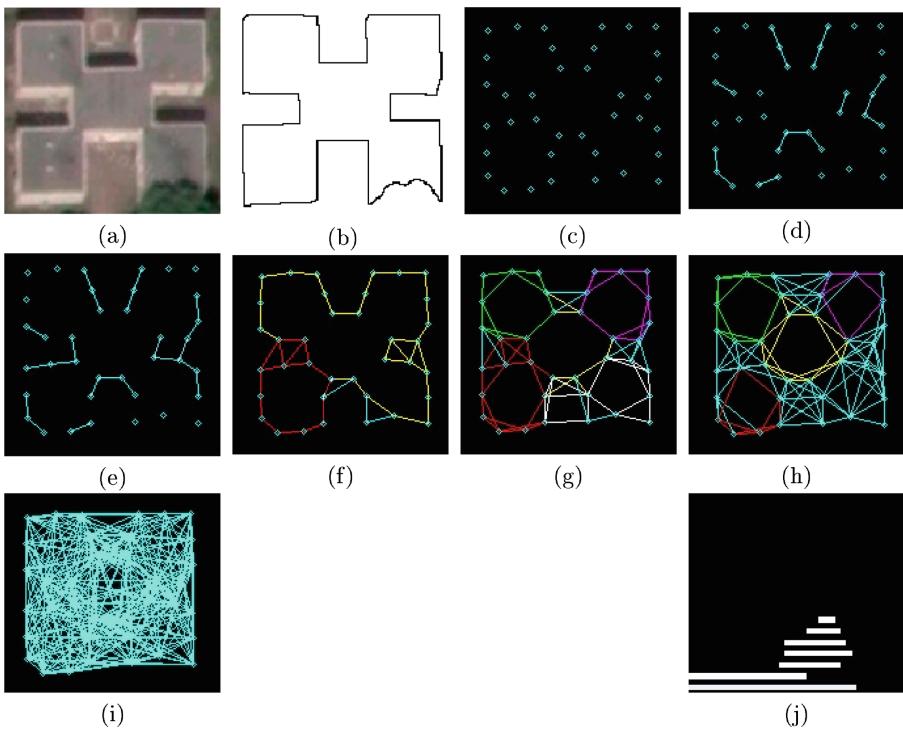


Fig. 2. (a) The source raster object, (b) the object contour, (c) the rarefied point space, (d, e, f, g, h, i) process of creation of simplicial complexes in case of different values $r(0; 80; 90; 130; 170; 210; 420)$ and (j) barcode.

The example of a process of formation of a simplicial complex K is presented in Fig. 2(c-i). After receiving rarefied point space (Fig. 2(c)) we begin to increase gradually radius and to connect points. We can watch formation of 8 components of connectivity in Fig. 2(d). In Fig. 2(e) the new component and one component received from two others are added. All components integrate in one in the case

of further increase in radius (Fig. 2(f)). Also, we watch formation of two holes. Six holes is formed in Fig. 2(g). Disappearance of some holes as a result of their partition on simplexes is fixed in Fig. 2(h). Figure 2(i) represents an object where all points are connected among themselves. The barcode of an object is shown in Fig. 2(j).

As a result, we receive the sequence of simplicial complexes

$$\emptyset \subset K_1 \subset K_2 \subset \dots \subset K_m.$$

Bettie's numbers $\beta_0(r)$ and $\beta_1(r)$ are applied to the computation of topological features. $\beta_0(r)$ is the number of components of connectivity on radius r , and $\beta_1(r)$ is the number of holes [18]. Each component of connectivity and hole have creation start time and end time: $d(\chi) - b(\chi)$, where χ is a life line of a topological feature.

The chart of persistence represents a set of all life lines which will form barcode

$$Dgm(V) = \{c_i\} (i = 1, 2, \dots, l),$$

where V is a source point set.

$$c_i = d(\chi_i) - b(\chi_i),$$

where c_i is the existence time of a component or hole, l is the number of components of connectivity or holes.

Stable features are shown on the big ranges of distances c_i of barcode $Dgm(V)$ and are a basis for the further analysis. The features which are shown at small distances c_i are the “noise” distorting idea of the layout of spatial objects and aren't subject to review.

3 Algorithm for Barcode Analysis of Spatial Objects

In this section, we present our algorithm for comparing of barcodes of spatial objects. It is an important step of our approach for establishing the correspondence of spatial objects on heterogeneous maps.

Source data are two sets of points of contours of the bitmap image $X = \{x_{\alpha_1}, x_{\alpha_2}, \dots, x_{\alpha_n}\}$ and $Y = \{y_{\alpha_1}, y_{\alpha_2}, \dots, y_{\alpha_m}\}$. The number of points n and m can be different. Besides, the number of points can be rather big therefore for an acceleration of operation of an algorithm we will use rarefied space of points. First, we will calculate key points which correspond to angles and characteristics of a contour. Then the remained points will be uniformly distributed on all space of points.

Barcodes $Dgm(X)$ and $Dgm(Y)$ are built on each set of points X and Y . We will consider barcodes for the analysis of holes. The algorithm works similarly for barcode analysis of components of connectivity. In fact, we receive two vectors of the topological features representing lines of the existence of each hole. The number of holes, i.e. capacities $|Dgm(X)|$ and $|Dgm(Y)|$ can be also various.

It is required to determine a level of similarity of topological features of two objects on the basis of data of a barcode. Long lines of existence correspond to steady characteristics and they play a key role in the case of identification of objects. We will consider that if length of lines there is smaller some given threshold, i.e. $c_X^i, c_Y^j < \varepsilon$ ($i \in 1, 2, \dots, l_X, j \in 1, 2, \dots, l_Y$), then they are noise and don't join in the analysis.

We will sort two vectors $Dgm(X)$ and $Dgm(Y)$ in decreasing order, i.e. $c_X^i \geq c_X^{i+1}$ ($i \in 1, 2, \dots, l_X - 1$) and $c_Y^j \geq c_Y^{j+1}$ ($j \in 1, 2, \dots, l_Y - 1$).

To allocate the significance of the first hole we normalize barcodes $Dgm(X)$ and $Dgm(Y)$ on the maximum element. We will find the greatest length among two barcodes $Dgm(X)$ and $Dgm(Y)$: $\max = \max(c_X^1, c_Y^1)$. If $c_X^1 < \max$, then $c_X^i = c_X^1 \cdot \frac{\max}{c_X^1}$ ($i = 2, 3, \dots, l_X$). If $c_Y^1 < \max$, then $c_Y^j = c_Y^1 \cdot \frac{\max}{c_Y^1}$ ($j = 2, 3, \dots, l_Y$).

Then we will calculate the number of all values of each vector:

$$S_X = \sum_{i=1}^{l_X} c_X^i, S_Y = \sum_{j=1}^{l_Y} c_Y^j.$$

We will determine the weight of each parameter from barcodes $Dgm(X)$ and $Dgm(Y)$ on the basis of total amount. It becomes the relation of value of each length of the line of existence of a hole in a barcode to the total amount of all parameters: $p = \{p_1, p_2, \dots, p_{l_X}\}$, where $p_i = \frac{c_X^i}{S_X}$ ($i = 1, 2, \dots, l_X$), and $t = \{t_1, t_2, \dots, t_{l_Y}\}$, where $t_j = \frac{c_Y^j}{S_Y}$ ($j = 1, 2, \dots, l_Y$).

To determine similarity of vectors $Dgm(X)$ and $Dgm(Y)$, we will calculate the relation of their elements which have identical indexes:

$$z = \{z_1, z_2, \dots, z_{\min(l_X, l_Y)}\},$$

$$\text{where } z_i = \begin{cases} \frac{c_X^i}{c_Y^i}, & \text{if } c_X^i < c_Y^i \\ \frac{c_Y^i}{c_X^i}, & \text{otherwise} \end{cases} \quad (i = 1, 2, \dots, \min(l_X, l_Y)).$$

In places where the index goes beyond amount of values of one of vectors $Dgm(X)$ and $Dgm(Y)$, parameters are considered as unlike and $z_{\min(l_X, l_Y)+1} = z_{\min(l_X, l_Y)+2} = \dots = z_{\max(l_X, l_Y)} = 0$.

Further, we will increase similarity vector z on a vector of weights p or t which is maximum on capacity (operation of multiplication) and we will add these results to calculate a similarity index Q :

$$Q = \begin{cases} \sum_{i=1}^{l_X} z_i \cdot p_i, & \text{if } l_X > l_Y \\ \sum_{i=1}^{l_Y} z_i \cdot t_i, & \text{otherwise} \end{cases} \quad (1)$$

Having these indices for all objects, we have an opportunity to allocate the most similar object on topology and to take it for an identical one.

Let Q_h is the index of similarity on holes and Q_c is the index of similarity on connectivity components. These indices are calculated on the Eq. 1. Their differences consist in source barcodes $Dgm(X)$ and $Dgm(Y)$. In the case with Q_h

they are formed by the analysis of holes, and in the case with Q_c are formed by the analysis of connectivity components.

Experiments have shown which barcode on holes has the most part of the percent of similarity in comparison with connectivity components. It can be expressed as follows w is the share of a barcode on holes, and $l - w$ is the share of a barcode on connectivity components. In the article, $w = 75$ experimentally is accepted, i.e. the barcode on holes gives a contribution of 75%, and the barcode on connectivity components gives 25%. The index of similarity Q_h is multiplied by 0.75, and the index of similarity Q_c is multiplied by 0.25. Thus, general index of similarity of $Q_s = Q_h + Q_c$. Completely similar objects will give $Q_s = 100$.

4 Results

4.1 Comparison of Objects in the Case of Map Generalization

The part of the river on multi-scale maps is considered as an example for the analysis of the operation of an algorithm. Contours of the river are shown in Fig. 3(a-d). Objects with a large number of details in contrast to Fig. 3(d) are shown in Fig. 3(a).

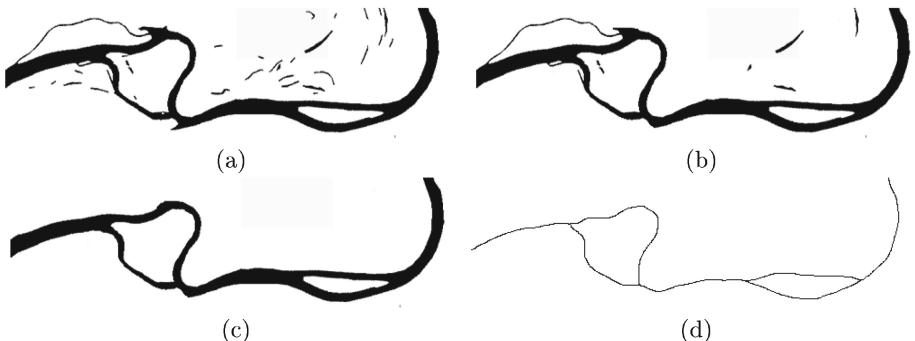


Fig. 3. Contours of maps of the rivers at different scales with different detailing.

We will build a barcode on the basis of a rarefied set of points (Fig. 4(a-d)).

Comparing barcodes (Fig. 6(a-c)), their visual similarity is noted though map objects at each new scale have fewer details in comparison with a previous one. Their form becomes more rough and angular. However, topological properties in the form of holes remain (Fig. 5(a-d)). The barcode in Fig. 6(d) differs from all the barcodes. This difference is caused by the strong simplification of the source map. The numerical characteristics of similarity calculated according to Eq. 1 are presented in Table 1 where a, b, c, d are objects presented in Fig. 3.

From Table 1 it is visible that indices of similarity of objects a, b, c have the close values. However, the object in Fig. 3(d) showed low results of similarity

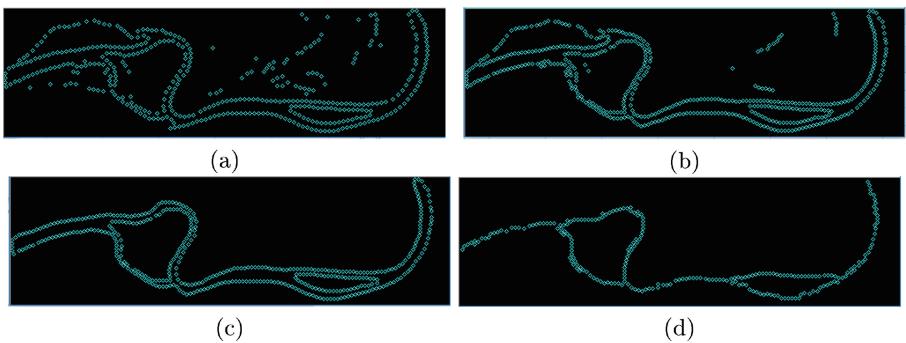


Fig. 4. Rarefied space of points of natural objects at different scales.

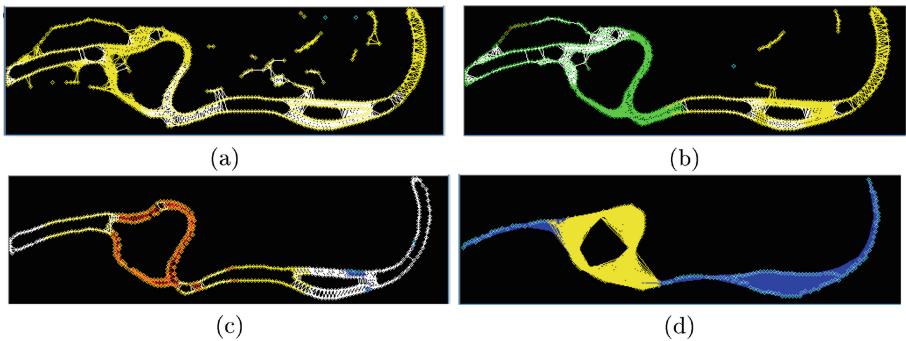


Fig. 5. Complexes in the course of formation of a barcode of natural objects at different scales at which it is possible to see the general structural elements (holes) visually.

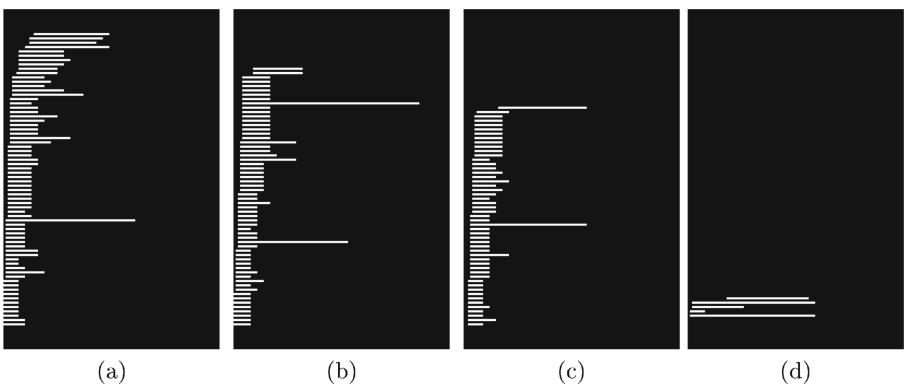


Fig. 6. Barcodes of holes of natural objects at different scales.

Table 1. Numerical characteristics of topological similarity of natural objects at different scales.

Rivers												
	a			b			c			d		
	Q_h	Q_c	Q_s	Q_h	Q_c	Q_s	Q_h	Q_c	Q_s	Q_h	Q_c	Q_s
a	75	25	100	49,53	21,18	70,71	51,08	18,23	69,31	23,74	18,37	42,11
b			75	25	100	57,43	20,82	78,25	27,42	20,98	48,40	
c						75	25	100	31,14	24,81	55,95	
d									75	25	100	

with other objects. In spite of the fact that all objects have similar main holes (Fig. 5(a-d)). The object on Fig. 3(d) has the minimum width of the bed of the river in difference from other objects. It is an important index because width of the bed of the river throughout all research forms a steady hole.

Total similarity and similarity on holes, on components of connectivity are specified for each object.

4.2 Comparison of Objects in the Case of Deformation of an Object

Comparison of objects in municipal GIS is complicated by the allocation of contours in connection with the existence of the superimposed objects on a map (Fig. 2(a)). We will execute deformation (Fig. 7(b)) and distortion (Fig. 7(c)) over the initial building (Fig. 7(a)). Further, we conduct similar researches, as in Sect. 4.1. Their results are shown in Figs. 8, 9 and 10.

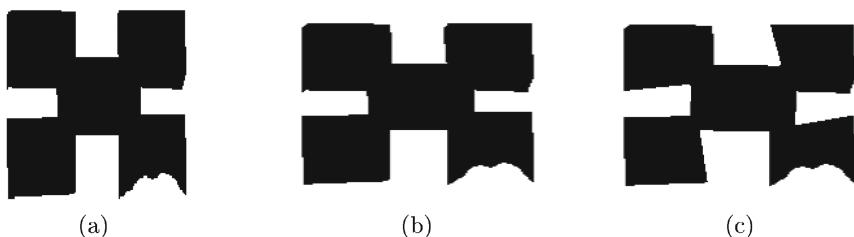


Fig. 7. Contours of municipal objects: (a) the source object, (b) the building with stretching, (c) the building with stretching and distortions.

We remove noise and we compare objects on barcodes without noise in contrast to maps of the rivers in the analysis of buildings (Table 2).

We see in Table 2 that an object *a* is similar to an object *c* stronger than to object *b*. It is connected with the fact that when stretching object *b* we increased the distance between touch points of two objects, i.e. reduced narrowings visually

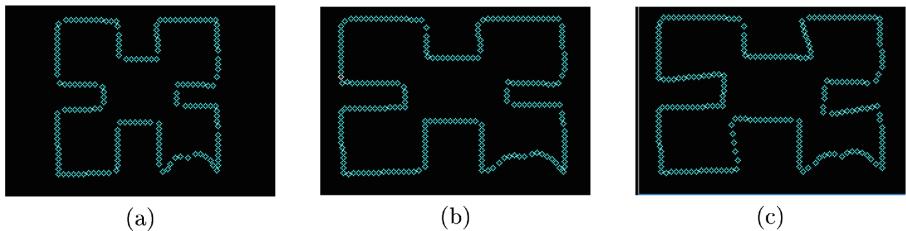


Fig. 8. Rarefied space of points of municipal objects: (a) the source object, (b) the building with stretching, (c) the building with stretching and distortions.

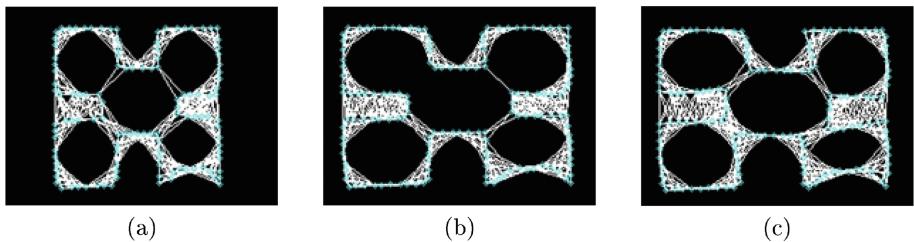


Fig. 9. Complexes in the course of formation of a barcode of municipal objects: (a) the source object, (b) the building with stretching, (c) the building with stretching and distortions.

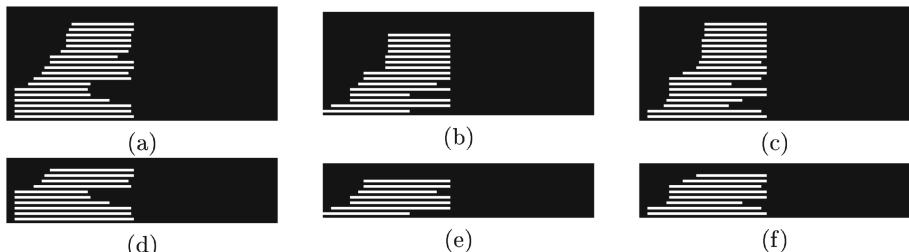


Fig. 10. Barcodes of holes with noise (a, b, c) and without noise (d, e, f) of municipal objects: (a, d) the source object, (b, e) the building with stretching, (c, f) the building with stretching and distortions.

Table 2. Numerical characteristics of topological similarity of municipal objects in the case of deformations and distortions.

City buildings										
	a			b			c			
	Q_h	Q_c	Q_s	Q_h	Q_c	Q_s	Q_h	Q_c	Q_s	
a	75	25	100	50,17	23,87	74,04	63,25	24,35	87,60	
b				75	25	100	60,33	24,50	84,83	
c							75	25	100	

dividing an object into several parts. As a result in the case of formation of holes the edge in this place turned out longer and was created in the case of a bigger radius. It influenced the later appearance of a hole.

It is necessary to note that as a result of deformation the bottleneck changed the form towards narrowing again. Though stretching was applied to an object *c* and the bottleneck was expanded. It led to the earlier appearance of a hole and prolonged time of its existence, then having made it more similar to object *a*.

The analysis of results shows the close similarity of barcodes of natural objects in the case of insignificant changes of detailing after generalization. However, the source barcode and the last one already strongly differ from each other in the case of a strong generalization. But the structure of an object remains, i.e. global characteristics of a source object match to global characteristics of an object after generalization in spite of the fact that detail information on one of the maps is absent. Also general topological features between buildings after deformation and distortions were revealed in the analysis of municipal objects.

5 Conclusion and Future Work

The task of comparison of objects on heterogeneous maps is considered in the article. The basis of an algorithm is the analysis of the form of objects which are exposed to deformations and distortions. However, the general structure of an object remains. Methods of persistent homology are invariant to such deformations. In order to compare spatial objects from different maps, their barcodes are analyzed. The algorithm for comparing of barcodes of two objects on raster maps is developed. The research of operation of an algorithm for different deformations of an object such as stretching an object and generalization of an object with saving the general structure is conducted. Results of experiments and comparison of spatial objects for natural and municipal maps are given.

Further, this approach can be used for accomplishing the following tasks. First, this automatic filling of attributive data of a map of one scale on the basis of data of a map of another scale. It will allow to carry out a quicker integration of spatial and semantic data of multi-scale maps. Also, an auto update of semantic information is urgent for maps of the same terrain for a different time frame. Another important direction of research in this field is an analysis of the video sequence received from the flight vehicle and comparison to the 3D model of terrain.

Acknowledgment. The reported study was funded by RFBR and Vladimir region according to the research project №17-47-330387.

References

1. Eremeev, S.V., Andrianov, D.E., Komkov, V.A.: Comparison of urban areas based on database of topological relationships in geoinformational systems. Pattern Recogn. Image Anal. **25**(2), 314–320 (2015)

2. Eremeev, S.V., Kuptsov, K.V., Andrianov, D.E.: Checking the topological consistency on maps of different scales. In: Supplementary Proceedings of the Fifth International Conference on Analysis of Images, Social Networks and Texts (AIST 2016), pp. 124–133 (2016)
3. De Almeida, J.P., Morley, J.G., Dowman, I.J.: A graph-based algorithm to define urban topology from unstructured geospatial data. *Int. J. Geogr. Inf. Sci.* **27**, 1514–1529 (2013)
4. Zhao, L., Peng, Q., Huang, B.: Shape matching algorithm based on shape contexts. *IET Comput. Vis.* **9**(5), 681–690 (2015)
5. Lomov, N.A., Mestetskiy, L.M.: Area of the disk cover as an image shape descriptor. *Comput. Opt.* **40**(4), 516–525 (2016)
6. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(10), 1615–1630 (2005)
7. Su, S., Ge, H., and Yuan, Y-H.: Multi-locality correlation feature learning for image recognition. In: IEEE Symposium on Computers and Communication (ISCC), pp. 874–879 (2016)
8. Carlsson, G., Zomorodian, A., Collins, A., Guibas, L.: Persistence barcodes for shapes. In: Proceedings of the 2004 Eurographics. ACM SIGGRAPH Symposium on Geometry Processing, pp. 124–135 (2004)
9. Skraba, P., Ovsjanikov, M., Chazal, F., Guibas, L.: Persistence-based segmentation of deformable shapes. In: Proceedings of CVPRW, pp. 45–52 (2010)
10. Su, Y., Liu, Y., Cuan, B., Zheng, N.: Contour guided hierarchical model for shape matching. In: IEEE International Conference on Computer Vision (ICCV), Santiago, pp. 1609–1617 (2015)
11. Ahmed, M., Fasy, B., Wenk, C.: Local persistent homology based distance between maps. In: SIGSPATIAL. ACM (2014)
12. Collins, A., Zomorodian, A., Carlsson, G., Guibas, L.: A barcode shape descriptor for curve point cloud data. *Comput. Graph.* **28**, 881–894 (2004)
13. Edelsbrunner, H.: Computational Topology: An Introduction. American Mathematical Society, Providence (2009)
14. Carlsson, E., Carlsson, G., de Silva, V., Fortune, S.: An algebraic topological method for feature identification. *Int. J. Comput. Geom. Appl.* **16**(4), 291–314 (2006)
15. Carlsson, G.: Topological pattern recognition for point cloud data. *Acta Numerica* **23**, 289–368 (2014)
16. Ghrist, R.: Barcodes: the persistent topology of data. *Bull. Am. Math. Soc.* **45**(1), 61–75 (2008)
17. Zhu, X.: Persistent homology: an introduction and a new text representation for natural language processing. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, pp. 1953–1959. AAAI Press (2013)
18. Edelsbrunner, H., Parsa, S.: On the computational complexity of Betti numbers: reductions from matrix rank. In: Proceedings of 25th ACM-SIAM Symposium on Discrete Algorithm, pp. 152–160 (2014)

Predicting Winning Team and Probabilistic Ratings in “Dota 2” and “Counter-Strike: Global Offensive” Video Games

Ilya Makarov^(✉), Dmitry Savostyanov, Boris Litvyakov,
and Dmitry I. Ignatov^{ID}

National Research University Higher School of Economics, Moscow, Russia
iamakarov@hse.ru

Abstract. In this paper, we present novel winning team predicting models and compare the accuracy of the obtained prediction with TrueSkill model of ranking individual players impact based on their impact in team victory for the two most popular online games: “Dota 2” and “Counter-Strike: Global Offensive”. In both cases, we present game analytics for predicting winning team based on game statistics and TrueSkill.

Keywords: Game analytics · Rating systems · TrueSkill
Machine learning · Data mining · Counter-Strike · Dota 2

1 Introduction

eSport is a rapidly growing direction having the advantage over traditional sports [1]. While some people still don't take it seriously, the viewership count records as well as prize pool records are being updated regularly during the biggest tournaments, reaching millions watching Dota 2 or Counter-Strike:GO. Both of the games are most popular games with over 500 000 humans playing simultaneously [2]. Such a popularity becomes a base for many betting companies oriented on eSport events [3].

It is worth mentioning, that eSport has a great advantage for game analytics over classic sport games providing structured information on the matches held online [4]. Many studies compared forecasting market with several rankings that appear not to be of high relevance to the real results [5, 6] due to limitations of information aggregation on the whole teams and bias of ratings with respect to different artificial measures. We aim to study on the quality of winning team prediction based on in-game analytics and individual rating systems. There are two aspects of sport analytics that we wanted to take into account in the current research: evaluation of players' ratings system for match making in online games and predicting the match outcome.

The task of assessing the level of the game of individual players in team eSport online games is of high practical importance. This work is focused on the popular eSport multiplayer online battle arena (MOBA) discipline Dota 2.

The aim of the work is to develop a method for ranking players of one team on the basis of personal contribution to the victory in the match. The Bayes formula and logistic regression are used as the core idea of the ranking system. The ratio of the estimated probabilities of team victory in the match based on the information about each single player of this team and on the information about the team as a whole. The result of the work is a model that allows estimating the player's contribution to the team victory using the basic game indicators and their dynamics throughout the match. In addition, the model makes it possible to compare the importance of factors influencing the victory, and can be used for match making system [7].

We then focus on a dynamic match result prediction based on the large dataset of demorecords for the core championships in multiplayer first-person shooter (FPS) called "Counter-Strike: Global Offensive". In fact, Counter-Strike is one the most popular shooters in the world for more than 10 years. It was originally developed in 1999 by Minh Le and Jess Cliffe as a Half-Life modification before the title rights moved to Valve. This paper describes a data-driven approach to identify game actions that lead to winning or losing in a game round after the bomb was planted on defense maps in Counter Strike.

Moreover, Bayesian rating model called TrueSkill is evaluated for both, "Dota 2" and "Counter Strike: GO" video games, in order to compare specific aspects of game analytics for different game genres similar to the comparison in [8].

2 Related Work

Several attempts were made to understand the key features of successful playing multiplayer first-person shooter online games [9, 10]. Most of the aspects under consideration were devoted to individual characteristics of human players, which sometimes are hard to measure [11]; moreover, these features may change over time. Several researchers try to evaluate statistic-based approaches of mining human behavior in FPS games [12, 13] and MOBA games [14–16].

In practical applications for online games, it is important to create a rating system for the problem of matchmaking, when the game should adapt team members in order to have the prior probability of a certain team winning as 50%. Fairness of such a system in Dota 2 game was evaluated in [17, 18], in general. For Counter-Strike we are the first to verify it. In what follows, we describe the individual and team ratings used in sport and eSport competitions.

2.1 Individual Ratings and Team Ratings

In many competitions, the organizers should compare players or teams while the players should be ranked in according to their results in the whole tournament. One of the first Bayesian rating system was the Elo rating developed for rating Chess tournament players [19]. The Elo rating was designed to provide unified ratings when there were no player who did not lose any match, which is the usual case in Chess tournaments. The idea for rating system could also be used

for matchmaking when we could see a battle of players with almost the same skill inspiring entertainment component of holding a competition.

Basically, the first ratings evolve under paradigm that one could compare several elements with respect to a certain number of simple properties, but could not compare all elements precisely and at once. For example, Bradley–Terry models [20] can be used for classification to multiple labels based on binary classification [21–24]. The comparison of the mentioned above Elo and BT ratings was presented in [25], while certain improvements of Elo system was published in [26, 27]. The application to sport ratings was presented in [28], in which the problem of learning rate over time was improved. Since the Elo rating invention, probabilistic rating systems have been generalized to handle team competitions with different team members between matches.

2.2 TrueSkill

The TrueSkill matching system was presented in [29]. TrueSkill is a Bayesian skill rating system which generalize the Elo rating used in Chess game [30, 31]. The presented system deals with an arbitrary number of competing teams and players. The main advantage of this system is that it can deduce individual skills from team results only. Despite it discards individual skills, the player is rated by the number of his impact on winning of his respective team. The system was evaluated in the “Halo 2” video game made by Microsoft [32].

The idea of this rating comes from ELO system that was adopted by many sports organizations around the world, including World Chess Federation FIDE [33]. The basic assumption of both rating systems is that players’ skill is a normally distributed random value. So players’ performance could slightly differ on different days but basic belief of the model is that it would be concentrated around some mean value. Two numbers, μ and σ , are used to describe the skill of each player: $skill_{team} \sim N(\mu, \sigma^2)$.

What differs TrueSkill from the ELO rating is that the former is adopted to work with any number of players in teams, including unequal teams. Another difference is that the ELO rating has a fixed value for the σ parameter while the TrueSkill algorithm generalizes the Elo rating by keeping track of two variables: the average skill μ and the system’s uncertainty about that estimate σ^2 [34]. These changes to the ELO system make TrueSkill more flexible, according to original research by Microsoft [29]. In matchmaking application for “HALO 2” video game, the lower bound $\mu - 3 \cdot \sigma$ was taken in order to stabilize skill learning policy for matching the players to opposing teams. Useful thing about both the ELO and TrueSkill rating models is that they allow to get winning probability for any two given teams in a direct way by simply subtracting two Gaussian random variables that stand for the opposing teams’ skill. They use Gaussian property that the sum/difference of two Gaussian random values is a Gaussian one: $P_1 \sim N(\mu_1, \sigma_1^2)$, $P_2 \sim N(\mu_2, \sigma_2^2)$, $P_{1-2} \sim N(\mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2)$.

Basic idea of modeling opposing teams’ performance is to see what are the chances that performance of one team would be better than the others. That is exactly what P_{1-2} here describes. So, in case its value is greater than zero,

the model assumption is that team1 will win. In other case the winner is team2, according to the model. It means that $P(P_{1-2} > 0) = \Phi\left(\frac{\mu_2 - \mu_1}{\sqrt{\sigma_1^2 + \sigma_2^2}}\right)$. Note that because there are always two opposing teams and no draws in both, Dota 2 and Counter-Strike games, two-teams case only is considered in this paper. However, it should be noted that TrueSkill could also provide draw chances and account any number of participants both in terms of teams and players.

The TrueSkill model was improved to handle ties (originally, omitted) [35] and different team sizes [36] with a training algorithm based on Expectation Propagation [37]. Later work experimented with additional information on tournament results, taking into account for the match-score differences [38, 39].

2.3 Machine Learning Ranking and Applications

A machine learning approach was used in order to evaluate player skill based on mouse-tracking technologies [40]. While such an approach seems promising, it could not be easily generalized for different games and require significant amount of time in order to catch the dynamic of game statistics during real eSport competitions. Neural networks were also used for predicting Bradley–Terry rating [41]. The rating systems are quite useful when considering the application to football ratings [42] and multi-label classification problems [43, 44]. The application of TrueSkill algorithm was suggested for context ads click prediction on the Web [45]. In order to test predicting winning team based on individual impact of team players, we use the implementation of Trueskill rating made in [46, 47].

2.4 TeamSkill

The research on individual and team performance in FPS games was made in [48]. In [49], the authors presented a prediction for winning team using the Elo, Glicko, and TrueSkill ratings, called TeamSkill. In what follows, the authors introduced several weighted approaches and included previously inseparable game-specific features improving prediction quality for the teams [50]. Their new TeamSkill-EVMixed classification method based on the threshold of the prior probability of defeat outperforms all previous approaches in tournament environments, even when a number of matches is small. The suggested approach was evaluated on Halo 3 video game history and NBA records [51]. However, in our work, we follow the standard definition of team score as a sum of team players skills, which is sufficient to get high accuracy of team ratings.

3 Probabilistic Rating Systems for “Dota 2” Video Game

3.1 “Dota 2” Overview

“Dota” is a multiplayer online battle arena video game in which two teams of five players try to destroy the opponent “Ancient” construction while defending their

own. The defense of the Ancients uses real-time strategy controls, represented on a single map with shadowed regions in 3D-isometric perspective.

Each player choose one of the 113 playable characters, known as “heroes”, each having their own advantages and drawbacks. Heroes are divided into two primary roles, known as the “carry” and “support”. Carries are the weakest ones on the start, but they are able to become the powerful ones, thus leading their team to the victory. Support heroes lack the abilities of making heavy damage; their purpose is to utilize the available resources to provide assistance for their respective carries. The two teams — called Radiant and Dire — appear at the fortified bases places in the opposite corners of the map, which itself is divided in half by a crossable river and connected by three paths, called “lanes” [52]. The lanes are guarded by defensive towers constantly attacking opposing units within their range. A small group of weak computer-controlled creatures called “creeps” goes through the predefined paths along the lanes and tries to attack any opposing units met on its way. Creeps spawn with some period from the two buildings, called the “barracks”, that exist in each lane and are located within on the team bases. The map is permanently covered for both teams in the “fog of war”, which does not allow a team from seeing the opposing team’s actions and units if they are not directly in sight of a player’s team unit.

4 Predicting Winning Team for “Dota 2”

The first stage in Dota 2 video game is also known as a pick stage or draft. Players from different teams alternately pick 10 heroes and ban 10 heroes which can be useful to the enemy side, or they do not want to play. Draft stage prediction was considered when building heroes recommender system made by Kevin Conley and Daniel Perry. Their k-Nearest Neighbors (kNN) model results in 70% of accuracy on test data which consists of 50,000 matches and 67.43% accuracy on overall data cross-validation; the logistic regression classifier trained 18,000 matches and got 69.8% accuracy on the test set [53]. The improvements of predictions on draft stage was made in [54].

It appears that some heroes in Dota 2 become more useful in tandem with some other heroes. In [55], the authors built the logistic regression model obtaining 62% prediction accuracy on the test set. It was caused by high dimensionality of model’s input vectors combined with the complexity of using hero statistics represented by principal components. In [56], another group of authors studied the tandem idea: they worked with history of 6,000 matches, representing feature space as 50 vectors of 2 hero interactions and obtained overfitted model with the accuracy on the training set 72%, and only 55% on the test data. The authors of [57] included interactions among the heroes and pairwise winning rate for Radiant and Dire teams. Despite of the overfitting problem the Random Forest and Logistic Regression algorithm demonstrated 67–73% accuracy on the test set. In [58], the authors made a comparison of heroes statistics via Logistic Regression, Support Vector Machines, Gradient Boosting, and Random Forest. The last one showed the best result and after some parameters tuning it was

used as a final classifier with 88.8% test accuracy showing that in-game process information makes a great boost of an estimator's accuracy.

In order to properly distribute the impact of each role in winning team, we should make a proper mapping between team players and their positions [59]. Understanding positions in a “Dota 2” team was solved by machine learning algorithms [60], and it is of great importance for predicting winning team [61].

4.1 Problem Setting

Consider a set of matches $M = \{m_1, \dots, m_n\}$ and a set of teams $T = \{t_1, \dots, t_k\}$. The match involves two teams and there are only two possible outcomes, i.e. one of the teams won. Each team contains 5 players. Let $P_t = \{p_1, \dots, p_5\}$ be a set of players of team $t \in T$. Teams do not change the composition of players, and the player can not be in several teams. Players in teams are assigned to roles $R = \{r_1, \dots, r_5\}$, also denoted as $R = \{\text{Carry, Mid-Lane Solo, Hard-Lane Solo, Semi-Support, Full Support}\}$. Each member of a team performs one role throughout a match, moreover, he performs the same role in all matches. Thus, there is a one-to-one correspondence between the set of players of the given team and the roles $\forall t \in T : P_t \longleftrightarrow R$. The problem is to rank the players of the team on the basis of personal contribution to the victory in this match.

4.2 Contribution Function

Because of one-to-one correspondence $P_t \longleftrightarrow R$ there is no difference between ranking of players or roles of the team in the match. Let us denote the probability of winning of a team t in a match m as $P_{m,t}(w)$. Note also that the roles forms a set of pairwise disjoint events whose union is the entire sample space. As a result, the total probability law can be used in the following way:

$$P_{m,t}(w) = \sum_{i=1}^5 P_{m,t}(r_i) \cdot P_{m,t}(w|r_i).$$

By definition, all roles are equally probable $\forall i, j : P_{m,t}(r_i) = P_{m,t}(r_j) = \frac{1}{5}$. So,

$$P_{m,t}(w) = \frac{1}{5} \cdot \sum_{i=1}^5 P_{m,t}(w|r_i), \text{ and } \sum_{i=1}^5 \frac{P_{m,t}(w|r_i)}{P_{m,t}(w)} = 5.$$

The roles' contribution to the victory in the current match is determined as

$$C_{m,t}(r_i) = \frac{P_{m,t}(w|r_i)}{P_{m,t}(w)}, \text{ with mean value } \sum_{i=1}^5 \frac{C_{m,t}(r_i)}{5} = 1$$

This function allows us to compare the players of a team t in a match m based on the contribution to the victory of the roles they perform.

4.3 Role-Based Model Evaluation

The probability $P_{m,t}(w|r_i)$ can be estimated by using logistic regression model. Every single role r_i in a match m for a team t can be represented as a vector $\mathbf{x} = \mathbf{x}(m, t, r_i) = (x_1, \dots, x_l)$. The vector consists of components which reflects such in-game information as “Gold Earned”, “Damage Dealt”, “Used Hero”, etc. It is allowed to estimate the probability of win based only on information about current role/player in the following way:

$$P_{m,t}(w|r_i) = \sigma(\langle \beta, \mathbf{x} \rangle),$$

where β is a vector of parameters, $\sigma(t) = \frac{1}{1+e^{-x}}$ is a logistic sigmoid.

4.4 Experiments

We consider professional competitions for “Dota 2” during March–April 2017 from Opendota resource [62], containing participant of Kiev Major grand challenge. We choose the following features from the open stats table: amount of gold and experience earned by each player on the end of the match, logs for kills, purchases, and ward placements; we have collected over 5000 records and 150 features. All the data is split into five roles, for which individual models are trained separately. 5-fold cross-validation quality metrics are represented in Table 1.

Table 1. Quality metrics for role-based models in “Dota 2”

Role	AUC	F1	Recall	Precision	Accuracy
r_1 - Carry	0.98	0.93	0.92	0.93	0.93
r_2 - Mid-Lane Solo	0.97	0.93	0.93	0.93	0.93
r_3 - Hard-Lane Solo	0.96	0.90	0.91	0.90	0.90
r_4 - Semi-Support	0.97	0.90	0.91	0.91	0.90
r_5 - Full Support	0.96	0.90	0.92	0.89	0.90

In addition, to evaluate predictions quality, aggregated team skill are computed as a sum of individual role impacts in winning team $P_{m,t}(w) = \sum_{i=1}^5 P_{m,t}(w|r_i)$. For every match m between teams t_1 and t_2 the following rule is used: if $P_{m,t_1}(w) > P_{m,t_2}(w)$ then t_1 won, else t_2 won. On the other hand, TrueSkill can be used as a baseline for predicting winning team in the middle of the match bounding from below 0.92 accuracy of prediction model by the 0.72 accuracy based on TrueSkill only.

4.5 Discussion on “Dota 2” Results

Without corresponding model of roles impact in Dota 2, we try to evaluate our approach not only by predicting a winning team, but also using an expert opinion represented below. Let us consider the match between OG and EG teams during Kiev Major tournament [63]. The role distribution is given by EG and OG teams.

In Table 2, the information on the end of the match for EG team is shown. We could see the players in positions 4 and 5 farm greater amount of gold than expected, while the relations between kills, deaths, and assists is better than for positions 1 and 3, meaning the lack of performance of the latter players. Player 2 performs well in terms of all the characteristics except damage to the opponent buildings, which should be one of the main priorities for positions 1 and 2, thus reducing performance of the second player. The described in-game analysis is well predicted, which is shown in Table 2 with a small drawback of decreased influence of role 1. As for OG team, players 4 and 5 performed well, but spent much gold to buy together 37 sentry wards, which reduces their impact for supporting computer-controlled players with lack of efficiency. Player 2 performed well in all the aspects except he died many times. Player 3 has less damage than player 5 while player 5 should be support class, thus reducing self-performance (Table 2). The lack of impact for player 1 who made 40% damage of the hole team can be described by the great number of deaths and the lack of damage to the opponent buildings.

Table 2. Estimated contribution to the victory

Team	Role	Player	Contribution	Team	Role	Player	Contribution
EG	Cr1t	5	0.35	OG	2	ana	0.28
	zai	4	0.34		4	JerAx	0.26
	Suma1L	2	0.26		5	Fly	0.23
	Universe	3	0.04		3	s4	0.15
	Arteezy	1	0.01		1	N0tail	0.08

5 Predicting Winning Team in Counter-Strike

5.1 Counter-Strike Overview

Counter-strike is a type of a game that is called tactical first-person shooter. The gameplay is based on shooting your opponents and trying to kill them looking from a first-person perspective. However, the game also provides a great diversity on its strategical and tactical parts because the competitive matches during world tournaments are played between two teams of five players each.

What differs tactical shooter from a DeathMatch or “Free-for-all” gaming mode is that a gaming location (called a map) has some specific goal. Achieving that goal leads to a victory in a round. Killing all of your opponents usually leads

to a win too, but sometimes the map goal can be reached even if everyone from a team was killed in a round. In competitive Counter-Strike maps, the goal of “Terrorists” team is to plant the bomb at some specific location called BombSite and prevent the Bomb from being defused. After the bomb is planted there is a 35–40 s countdown. The opposing team called “Counter-terrorists” aims to prevent bomb planting or defuse it in a limited amount of time after plant, otherwise they lose a round after the bomb explodes. In both cases, the way of winning by killing all the players from the opposite team is also a way to win the round. Teams play rounds repetitively until the end of the 15th round when the teams switch sides and continue playing until one of the teams would have 16 rounds won in total. In case of 15–15 score a few additional rounds can be played to define a winner if needed. We consider the model of predicting a winning team based on after-plant in-game situations. After the bomb is planted, the 40-s countdown is started. All of the situations are split into 1-s time intervals, for which we try to measure the winning chances.

In order to build prediction model we, first, have downloaded game replays, which are further used for loading game attributes with the self-made parser based on OpenSource project by StatsHelix [64] and have extracted raw game data into a .csv file. Using Google’s Protocol Buffers as a message/object serialization language we parse the original .dem files for the game records from the world changes in a sequential way [65]. The dataset covers the last four years of demorecords from the storage [66]. A C# wrapper is built over the demoinfo tool to get the raw data regarding game events, such as: kills, shots, movements, player coordinates and view directions, and several descriptive statistics over them. Most of the chosen features are related to after-plant situations with respect to the round goal, i.e. to explode/defuse the planted bomb, depending on a team: the number of players alive, difference in the current team sizes, the total and average equipment cost, the number of damaged and healthy players, the smoke cover on bomb plant, the total number of different grenades, the total TS of a team, and TS prediction on a winner. 162 demos in total have been harvested to feed the after-plants model. It was noticed that most of these games were played during the period October 2016–January 2017.

6 Experiments

6.1 Metrics Used

We use TrueSkill judged based on accuracy and Log-Loss, while metrics for prediction of winning team with after-plant feature analysis using Decision Trees and Logistic Regression were taken as accuracy and Log-Loss. In case of binary classification problem one could imagine a naive prediction model of a fair coin tossed for every prediction, which should be worse than god prediction model. For a fair coin toss, Log-Loss on average is close to 0.7, while average accuracy is exactly 50%.

Table 3. Comparing ratings

data_train	data_test	acc	logloss
all games	all games	0.62	0.675
dust2 only	dust2 only	0.59	0.69
all games	dust2 only	0.57	0.75

6.2 TrueSkill for Winning Team Prediction in Counter-Strike

Each player is given with a personal rating. We go through the list of played games and refresh the ratings after each game. Different pre-learn periods are tested on a test sample of 6 different months. 9 months period is chosen for maximizing accuracy or the prediction models, which stabilizes on 0.68 value with 0.61 Log-Loss change. We choose learning TrueSkill from 2016-02 until 2016-09, with just 3 games from a set played in September, placing a gap before the test dataset. The prediction period chosen is in between 2016-09-28 and 2017-03-11. Three combinations of train/test datasets based on all the games or de_dust_2 only games were used during evaluation. The results are shown in the Table 3.

7 Discussion and Future Work

We have considered the application of machine learning techniques for predicting winning team for the two most popular online games. The results obtained show that the quality of prediction is higher when we check the in-game parameters closer to the round end. We use TrueSkill rating system to measure baseline for the prediction model when we want to step back from the end of the match and have a prediction on game features that should not have less accuracy than prediction based on a Bayesian probabilistic rating system. We are looking forward to compare our system with other rating systems and improve TrueSkill model [39] in order to take into account the role impact distribution during the game round for both video games.

Acknowledgments. The work was supported by the Russian Science Foundation under grant 17-11-01294 and performed at National Research University Higher School of Economics, Russia. We would like to thank Alexander Semenov and Petr Romov for their piece of advice.

References

1. Taylor, T.: *Raising the Stakes: E-sports and the Professionalization of Computer Gaming*. MIT Press, New York (2012)
2. Powered by Steam: Steamcharts. An ongoing analysis of steam's concurrent players (2017). <http://steamcharts.com/>. Accessed 09 May 2017

3. Kaytoue, M., et al.: Watch me playing, i am a professional: a first study on video game live streaming. In: Proceedings of the 21st International Conference on WWW, NY, USA, pp. 1181–1188. ACM (2012)
4. Wagner, M.G.: On the scientific relevance of eSports. In: International Conference on Internet Computing, pp. 437–442 (2006)
5. Luckner, S., Schröder, J., Slamka, C.: On the forecast accuracy of sports prediction markets. In: Gimpel, H., Jennings, N.R., Kersten, G.E., Ockenfels, A., Weinhardt, C. (eds.) Negotiation, Auctions, and Market Engineering. LNBIP, vol. 2, pp. 227–234. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-77554-6_17
6. Tsai, M.: Fantasy (e)Sports: the future prospect of fantasy sports betting amongst organized multiplayer video game competitions. UNLV Gaming LJ **6**, 393 (2015)
7. Zhang, L., et al.: A factor-based model for context-sensitive skill rating systems. In: 2010 22nd IEEE International Symposium on TAI, vol. 2, pp. 249–255 (2010)
8. Coulom, R.: Whole-history rating: a Bayesian rating system for players of time-varying strength. In: van den Herik, H.J., Xu, X., Ma, Z., Winands, M.H.M. (eds.) CG 2008. LNCS, vol. 5131, pp. 113–124. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87608-3_11
9. Dick, M., Wellnitz, O., Wolf, L.: Analysis of factors affecting players' performance and perception in multiplayer games. In: Proceedings of 4th ACM SIGCOMM IW on Network and System Support for Games, NY, USA, pp. 1–7. ACM (2005)
10. Wright, T., Boria, E., Breidenbach, P.: Creative player actions in FPS online video games: playing counter-strike. Game Stud. **2**(2), 103–123 (2002)
11. Rioult, F., Métivier, J.P., Helleu, B., Scelles, N., Durand, C.: Mining tracks of competitive video games. AASRI Procedia **8**, 82–87 (2014)
12. Hladky, S., Bulitko, V.: An evaluation of models for predicting opponent positions in first-person shooter video games. In: 2008 IEEE International Symposium on CIG, pp. 39–46 (2008)
13. Bird, A.M.: Development of a model for predicting team performance. Am. Alliance Health Phys. Educ. Recreat. **48**(1), 24–32 (1977)
14. Drachen, A., et al.: Skill-based differences in spatio-temporal team behaviour in defence of the ancients 2 (dota 2). In: 2014 IEEE GME, pp. 1–8 (2014)
15. Pobiedina, N., et al.: On successful team formation: Statistical analysis of a multiplayer online game. In: 2013 IEEE 15th International Conference on Business Informatics, pp. 55–62 (2013)
16. Yang, P., Roberts, D.L.: Knowledge discovery for characterizing team success or failure in (A)RTS games. In: 2013 IEEE International Conference on CIG, pp. 1–8, August 2013
17. Wu, M., Xiong, S., Iida, H.: Fairness mechanism in multiplayer online battle arena games. In: Proceedings of 3rd International Conference on SAI (ICSAI), pp. 387–392, November 2016
18. Myślak, M., Deja, D.: Developing game-structure sensitive matchmaking system for massive-multiplayer online games. In: Aiello, L.M., McFarland, D. (eds.) SocInfo 2014. LNCS, vol. 8852, pp. 200–208. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15168-7_25
19. Elo, A.: The Rating of Chessplayers, Past and Present. Arco Pub., New York (1978)
20. Bradley, R.A., Terry, M.E.: Rank analysis of incomplete block designs: I. the method of paired comparisons. Biometrika **39**(3/4), 324–345 (1952)
21. Wu, T.F., Lin, C.J., Weng, R.C.: Probability estimates for multi-class classification by pairwise coupling. J. Mach. Learn. Res. **5**, 975–1005 (2004)
22. Huang, T.K., et al.: Generalized Bradley-Terry models and multi-class probability estimates. J. Mach. Learn. Res. **7**, 85–115 (2006)

23. Fujimoto, Y., Hino, H., Murata, N.: An estimation of generalized Bradley-Terry models based on the em algorithm. *Neural Comput.* **23**(6), 1623–1659 (2011)
24. Matsumoto, I., et al.: Online density estimation of Bradley-Terry models. In: Proceedings of International Conference on Learning Theory, Paris, France, pp. 1343–1359. PMLR (2015)
25. Király, F.J., Qian, Z.: Modelling Competitive Sports: Bradley-Terry-Elo Models for Supervised and On-Line Learning of Paired Competition Outcomes. arXiv preprint [arXiv:1701.08055](https://arxiv.org/abs/1701.08055) (2017)
26. Glickman, M.E.: The Qlicko System. Boston University, Boston (1995)
27. Glickman, M.E.: Example of the Qlicko-2 System. Boston University, Boston (2012)
28. Glickman, M.E., Hennessy, J., Bent, A.: A comparison of rating systems for competitive women's beach volleyball. <http://www.glicko.net/>
29. Herbrich, R., Minka, T., Graepel, T.: Trueskill™: a Bayesian skill rating system. In: Proceedings of the 19th International Conference on NIPS, MA, USA, pp. 569–576. MIT Press (2006)
30. Graepel, T., Herbrich, R.: Ranking and matchmaking. *Game Dev. Mag.* **25**, 34 (2006)
31. Dangauthier, P., Herbrich, R., Minka, T., Graepel, T., et al.: Trueskill through time: revisiting the history of chess. In: NIPS, pp. 337–344 (2007)
32. Huang, J., et al.: Mastering the art of war: how patterns of gameplay influence skill in halo. In: Proceedings of the SIGCHI International Conference, NY, USA, pp. 695–704. ACM (2013)
33. Wikipedia: Fide world rankings - wikipedia, the free encyclopedia (2017). https://en.wikipedia.org/w/index.php?title=FIDE_World_Rankings&oldid=776755738. Accessed 5 May 2017
34. Moser, J.: Computing your skill (2010). <http://www.moserware.com/2010/03/computing-your-skill.html>. Accessed 9 May 2017
35. Nikolenko, S., Sirotkin, A.: A new Bayesian rating system for team competitions. In: Proceedings of the 28th International Conference on Machine Learning (ICML 2011), pp. 601–608 (2011)
36. Nikolenko, S.I., Sirotkin, A.V.: Extensions of the trueskill™ rating system. In: Proceedings of the 9th International Conference on AFSSC, pp. 151–160. Citeseer (2010)
37. Bishop, C.M.: Pattern recognition. *Mach. Learn.* **128**, 1–58 (2006)
38. Nikolenko, S.I., Serdyuk, D.V., Sirotkin, A.V.: Bayesian rating systems with additional information on tournament results. *Trudy SPIIRAN* **22**, 189–204 (2012)
39. Nikolenko, S.: A probabilistic rating system for team competitions with individual contributions. In: Khachay, M.Y., Konstantinova, N., Panchenko, A., Ignatov, D.I., Labunets, V.G. (eds.) AIST 2015. CCIS, vol. 542, pp. 3–13. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26123-2_1
40. Buckley, D., Chen, K., Knowles, J.: Rapid skill capture in a first-person shooter. *IEEE Trans. Comput. Intell. AI Games* **9**(1), 63–75 (2017)
41. Menke, J.E., Martinez, T.R.: A Bradley-Terry artificial neural network model for individual ratings in group competitions. *Neural Comput Appl.* **17**(2), 175–186 (2008)
42. Tarlow, D., Graepel, T., Minka, T.: Knowing what we don't know in NCAA football ratings: understanding and using structured uncertainty. In: Proceedings of the 2014 MIT Sloan Sports Analytics Conference (SSAC 2014), pp. 1–8. Citeseer (2014)
43. Lee, J.-S.: TrueSkill-Based pairwise coupling for multi-class classification. In: Villa, A.E.P., Duch, W., Érdi, P., Masulli, F., Palm, G. (eds.) ICANN 2012. LNCS,

- vol. 7553, pp. 213–220. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33266-1_27
- 44. Naik, N., et al.: Streetscore-predicting the perceived safety of one million streetscapes. In: Proceedings of the IEEE International Conference on CVPR Workshops, pp. 779–785 (2014)
 - 45. Graepel, T., et al.: Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In: Proceedings of the 27th International Conference on Machine Learning (ICML 2010), pp. 13–20 (2010)
 - 46. Hamilton, S.: Pythonskills: implementation of the trueskill, glicko and elo ranking algorithms (2012)
 - 47. Lee, H.: Python implementation of trueskill: the video game rating system (2013)
 - 48. Shim, K.J., et al.: An exploratory study of player and team performance in multiplayer first-person-shooter games. In: 2011 IEEE 3rd International Conference on Privacy, Security, Risk and Trust and 3rd International Conference on Social Computing, pp. 617–620, October 2011
 - 49. DeLong, C., Pathak, N., Erickson, K., Perrino, E., Shim, K., Srivastava, J.: TeamSkill: modeling team chemistry in online multi-player games. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011. LNCS (LNAI), vol. 6635, pp. 519–531. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20847-8_43
 - 50. DeLong, C., Srivastava, J.: TeamSkill evolved: mixed classification schemes for team-based multi-player games. In: Tan, P.-N., Chawla, S., Ho, C.K., Bailey, J. (eds.) PAKDD 2012. LNCS (LNAI), vol. 7301, pp. 26–37. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30217-6_3
 - 51. DeLong, C., Terveen, L., Srivastava, J.: TeamSkill and the NBA: applying lessons from virtual worlds to the real-world. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in SNA and Mining, NY, USA, pp. 156–161. ACM (2013)
 - 52. McDonald, T.: A beginner's guide to dota 2: Part one - the basics (2013). <https://www.pcinvasion.com/a-beginners-guide-to-dota-2-part-one-the-basics>. Accessed 25 July 2013
 - 53. Conley, K., Perry, D.: How does he saw me? A recommendation engine for picking heroes in dota 2. Np, nd Web 7 (2013)
 - 54. Semenov, A., Romov, P., Korolev, S., Yashkov, D., Neklyudov, K.: Performance of machine learning algorithms in predicting game outcome from drafts in dota 2. In: Ignatov, D.I., Khachay, M.Y., Labunets, V.G., Loukachevitch, N., Nikolenko, S.I., Panchenko, A., Savchenko, A.V., Vorontsov, K. (eds.) AIST 2016. CCIS, vol. 661, pp. 26–37. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-52920-2_3
 - 55. Agarwala, A., Pearce, M.: Learning dota 2 team compositions. Technical report, Stanford University (2014)
 - 56. Song, K., Zhang, T., Ma, C.: Predicting the winning side of dota2. Technical report, Stanford University (2015)
 - 57. Yang, Y., Qin, T., Lei, Y.H.: Real-time esports match result prediction. arXiv preprint [arXiv:1701.03162](https://arxiv.org/abs/1701.03162) (2016)
 - 58. Johansson, F., Wikström, J.: Result prediction by mining replays in dota 2 (2015)
 - 59. Inkarnate: Dota 1-to-5 system (2012). <http://www.liquiddota.com/forum/dota-2-strategy/454943-dota-1-to-5-system>. Accessed 05 Sep 2011
 - 60. Eggert, C., Herrlich, M., Smeddinck, J., Malaka, R.: Classification of player roles in the team-based multi-player game dota 2. In: Chorianopoulos, K., Divitini, M., Hauge, J.B., Jaccheri, L., Malaka, R. (eds.) ICEC 2015. LNCS, vol. 9353, pp. 112–125. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24589-8_9

61. Pobiedina, N., et al.: Ranking factors of team success. In: Proceedings of the 22 International Conference on World Wide Web, NY, USA, pp. 1185–1194. ACM (2013)
62. Powered by Steam: Stats from professional dota 2 matches (2017). <https://www.opendota.com/explorer>. Accessed 01 May 2017
63. DotaBuff: Kiev major: team eg vs. team og (2017). <https://www.dotabuff.com/matches/3148721353>. Accessed 16 May 2017
64. StatsHelix: Cs:go demos parser by statshelix (2014). <https://github.com/StatsHelix/demoinfo>. Accessed 9 May 2017
65. Valve: csgo-demoinfo (2014). <https://github.com/ValveSoftware/csgo-demoinfo/tree/master/demoinfogo>. Accessed 5 May 2017
66. HLTV.org: Hltv.org demos section (2017). hltv.org/?pageid=28. Accessed 9 May 2017



Bagging Prediction for Censored Data: Application for Theatre Demand

Evgeniy M. Ozhegov^(✉) and Alina Ozhegova

Group for Applied Markets and Enterprises Studies,
National Research University Higher School of Economics, Moscow, Russia
tos600@gmail.com

Abstract. In this research we analyze the demand for performing arts. Since the observed demand is limited by the capacity of house, one needs to account for demand censorship. The presence of consumer segments with different purposes of going to the theatre and willingness-to-pay for performance and ticket characteristics compels to account for heterogeneity in theatre demand. In this paper we propose an estimator for prediction of demand that accounts for both demand censorship and preferences heterogeneity. The estimator is based on the idea of classification and regression trees and bagging prediction aggregation. We extend the algorithm for censored data prediction problem. Our algorithm predicts and combines predictions from both discrete and continuous parts of censored data. We show that the estimator is better in prediction accuracy compared with estimators which account for censorship or heterogeneity of preferences only.

Keywords: Theatre demand · Bagging · Censored data

1 Introduction

Currently, firms, households and society as a whole generate, collect and store enormous volume of data starting from the level of individuals to the level of countries. This kind of data can be beneficial for forecasting of social development and behavior of economic agents, and evaluation of state-run programs. Availability of large data sets has made it possible to solve forecasting problems by applying methods of machine learning.

Machine learning methods divide the prediction problem into problems of variables and models selection. There are two common types of variables to be predicted. Categorical variables express belonging of an object to a certain class from discrete set. Continuous variables reflect a quantitative measure of object state. Meanwhile, models of economic predictions require statistical methods dealing with discrete-continuous variables such as censored or limited dependent

The original version of this paper was revised: An acknowledgement has been added. The Erratum to the paper is available at https://doi.org/10.1007/978-3-319-73013-4_38

variables. Censored data often arise in the models of individual consumption, where consumers either do not demand the good (zero consumption) or demonstrate positive amount of consumption. Since the consumption is left censored by zero, the data consist of discrete (choice of zero or non-zero consumption) and continuous (choice of amount when consumption is non-zero) components. Models of product demand with limited capacity also suffer from the problem of censored data. Since the seller cannot supply a good over particular amount per unit time, the demand is right-censored by maximum stock level. In such a case, the potential demand may exceed the observed demand, the amount of good that the consumers are willing to purchase and the seller is capable to supply. In the former case ignoring censored nature of data results in biased prediction. Model calibration on uncensored observations allows to estimate the change in consumption correctly but fails to predict transition to the group of consumers with zero consumption. In relation to the latter, methods ignoring the fact of censorship lead to underestimated effects and biased prediction of consumption. Inaccurate estimation causes nonoptimal pricing and loss of expected gain.

Within the context of demand estimation, it is crucial to account for demand heterogeneity, that arises from differentiated goods with a variety of characteristics and consumers with different preferences. Model of demand that does not take into account customer and product heterogeneity tends to estimate the effects and predict the consumption for an averaged good. Modelling the heterogeneity allows to detect the differences in customer preferences towards good characteristics, to reveal willingness-to-pay for different goods and to adapt pricing policy to certain product and consumer segments.

Econometric methods applied researchers of demand progress in consistent estimation of regressions on censored data. Traditional methods of limited dependent variable (LDV) estimation (Tobin 1958; Heckman 1977) are based on distributional assumptions of dependent variable or error term. This approach is sensitive to the choice of distributional assumption. At the same time, the lack of tests on assumption validity limits the accuracy of results. Modern nonparametric extensions of LDV models (Das *et al.* 2003; Matzkin 2012) relax distributional assumptions. However, nonparametric estimation with several independent variables leads to computational burden and slow rate of convergence that result in practical limitation on the number of explanatory variables and partial linearizing of a model. Semiparametric approach of censored quantile regression (Chernozhukov and Hong 2002; Chernozhukov *et al.* 2015) also allows to model demand on censored data without distributional assumption. Model estimation on different levels of quantile is a convenient way to account for heterogeneity of effects. Meanwhile, this approach is not suitable for prediction goals, since it requires the value of quantile for estimation of effects, that is unobservable in out-of-sample data.

Modern methods of machine learning as well as nonparametric models are based on the principle of model construction that would be optimal on some criterion in each data subspace. ML methods assume heterogeneity of objects and that source of heterogeneity is either unknown or unobserved for modeler. The core of subject consists of partitioning the characteristic space into a series

of hyper-cubes and model calibration for each of those partitions. One of the approaches based on the principle is classification and regression trees (CART) (Breiman *et al.* 1984). CART and its extensions (Breiman 1996; 2001) are gradually spread among econometricians and even now are widely used in prediction models of heterogeneous demand (Bajari *et al.* 2015). Machine learning methods prove its worth in prediction models with heterogeneous objects, since they do not require *a priori* assumptions on sources of heterogeneity. At the same time, these methods possess higher rate of convergence compared to nonparametric models that make them highly sought when data sets contain large number of predictors.

In this research we adapt ML prediction methods to censored data. The principles of machine learning methods can be carry over on calibration process of censored quantile regression. Combination of regression trees and censored quantile regression approaches allows to develop an algorithm for limited dependent variable prediction. This approach does not rely on *a priori* distributional assumptions as well as assumptions on sources of heterogeneity. Higher rate of convergence permits to estimate the models with a huge number of explanatory variable, that is complicated with nonparametric models. The algorithm consists of three steps: (1) Bagging prediction of dummy whether the dependent variable is on the censoring bound using classification trees; (2) Bagging prediction of dependent variable for observations classified as uncensored using median regression trees; (3) Trimming of second-step prediction and its combination with first-step prediction of censored observations.

We apply the method to a demand estimation problem. We use data on Perm Opera and Ballet tickets sales data that cover all performances for four seasons between August 2011 and July 2015 and include information on ticket purchase and performance characteristics. Structure of data disaggregated to the level of particular pricing area in a house allows to control on quality of seat as well. We use performance (production type, composer, band director etc.), play (month, day of a week, time of a day, premiere play) and seat (seating area dummies) characteristics to predict attendance rate and study variables importance. The unit of observation presents demand (attendance rate) for a particular seating area in a house on a particular performance. Since the prediction of demand presents an important problem for theatre management, the results of this research allow to propose recommendations upon price differentiation over seats and performances.

We find better performance of our algorithm in terms of predictive power compared with simple parametric methods (OLS and median regression), parametric methods (Tobit model and censored quantile regression) which accounts for data censorship but not for heterogeneity and method (tree of median regressions) that accounts for heterogeneity only. We study the mean structure of a tree and find that the most frequent variables for splitting the sample on subsamples with different effects are type of production (ballet or opera), seating area, nationality of composer (Russian or foreign), world fame of production and band director. We estimate the distribution of price effect comparing the predictions of attendance rate with current prices and prices increased by 10%. The

estimated price elasticity varies from 0 to -0.30 with a median equal to -0.07 that indicates on weakly elastic demand. We find that price elasticity of demand varies substantially with less elastic demand for ballets, Russian ballets among ballets and foreign operas among operas, and seats in the center of stalls.

2 Data

The data for research are taken from the Perm Opera and Ballet Theatre, which is considered as one the best regional opera theatre in Russia. It is famous for its modern musical productions, nonstandard classical performances, and unconventional festival projects. It is also a major Russian center for opera and ballet, where the quality of the musical performance is paramount. Every year the theatre performs forty regular productions and three to five new productions. The Perm Opera and Ballet Theatre is a non-commercial organization and as such is loss-making. Its main source of funding is a Perm state budget. As a non-commercial venture the goal of the theatre is to make ballet and symphonic art available for Perm residents. The theatre does have to, at least partially, recoup the expenses with production revenue in order to produce new ones. Consequently, the theatre constantly tries to balance between being affordable and covering costs using pricing mechanism and charging different prices for different performances and seats.

The data collected cover all performances for four seasons between August 2011 and July 2015. There were 298 performances out of 36 repertoire productions at the main venue. The data include information on the name of production, the date and time of play (season, year, month, the day of week and time of day), the price of a ticket, time and date of ticket purchase and the location of a seat in a house. The house of the theatre is divided into sectors: loges, the stalls, tiered stalls, the circle and the upper circle. In the sectors, the seats are identified by row and place. Further, the house is divided into nine seating areas according to the distance from the stage (Fig. 1). The seats in different areas vary by the quality of view and sound, prestige and price. Whereas the seats located in one area are considered as homogeneous in terms of price and quality.

In addition to the information provided by the theatre, we collect information on performance characteristics which explains the demand according to previous research (Corning and Levy 2002; Seaman 2006). We classify productions into operas and ballets, into classical (written before 1900) and modern (written after 1900) ones. We collect information on the composer and construct dummy responsible for the nationality of the composer (Russian/foreign) and the dummy on whether the production is a premiere one. We classified performances according to the age recommended for attendance: children (without restriction), family (12+) and adult (16+). Information on conductors allows estimating the contribution of a particular person. Among conductors, we identified two persons that are especially successful and in-demand. Perm Opera and Ballet Theatre has been regularly nominated for the prestigious Russian theatre award “Golden Mask”. For each production, we collect information on the number of nominations and awards won. In order to measure the world popularity

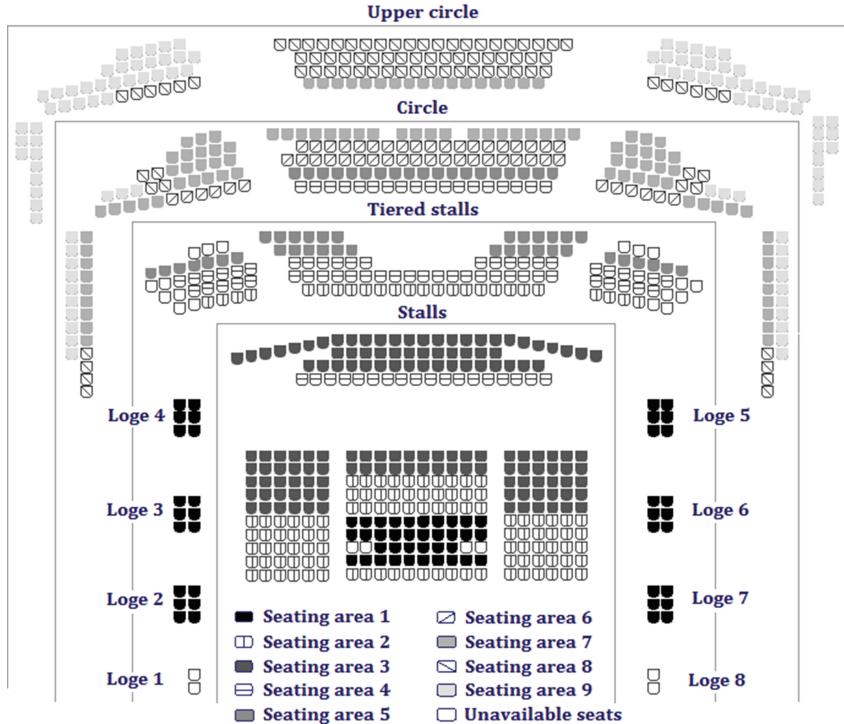


Fig. 1. The scheme of the house

of musical composition, we collect the data on various ratings. We use data from the worldwide rating of operas and their composers (operabase.com) and of ballets (listverse.com). Descriptive statistics of performances characteristics are presented in Table 1.

To estimate the model of demand, we aggregate data on sales and prices by seating areas. For each seating area we calculate the attendance rate as a number of sold tickets to the total number of seats in the area and assign the basic price in accordance with one of 8 theatre pricing schemes. The pricing scheme is the set of prices for 9 seating areas. Prices for the most expensive tickets (the first seating area) vary from 300 to 2000 rubles while the cheapest tickets (the ninth seating area) are always sold for 100 rubles.

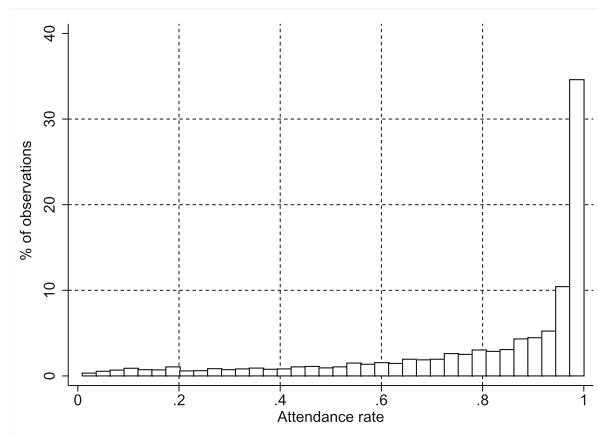
Apart from the seats in the house, the productions may also be heterogeneous. Figure 2 shows that half of the observations are filled over 80%. The remaining seating areas show lower demand which tells us about the heterogeneity of productions.

One more issue to be discussed is a potentially different quality of seats for different types of productions. Seats are heterogeneous in terms of view and sound quality which are not ordered strictly according to seating area number (and price of a ticket). Thus, seats closer to the stage may be not the best

Table 1. Descriptive statistics

Variable	Total	Share
<i>Day of week</i>	2682	
Working days	1440	46.3
Weekend	1242	53.7
<i>Time of day</i>	2682	
Before 2 am	342	12.8
After 2 am	2340	87.2
<i>Type of performance</i>	2682	
Ballet	954	35.6
Opera	1728	64.4
<i>World rating of performance</i>	2682	
Rated	1017	37.9
Not rated	1665	62.1
<i>Language of opera</i>	2682	
Foreign	378	14.1
Russian	2304	85.9
<i>Recommended age</i>	2682	
Without restrictions	1107	41.3
From 12 y.o	1170	43.6
From 16 y.o	405	15.1
<i>Awards</i>	2682	
Presence	144	5.4
Absence	2538	94.6
<i>The nationality of composer</i>	2682	
Russian	1521	56.7
Foreign	1161	43.3
<i>Band director</i>	2682	
Valeriy Platonov	1494	55.7
Teodor Currentzis	279	10.4
Others	909	33.9

to watch a ballet since the level of stalls is lower than the level of the stage. Theatre experts' opinion is that the best seats for watching a ballet are located in the center of circle which corresponds to fourth to sixth seating areas. This is supported by the data on attendance of performances and seats disaggregated by production type (Table 2). The most filled areas at ballets are areas 4–7 while for operas the most filled areas are 2–4. This corresponds to a higher quality of sound in this areas and the higher importance of sound quality in operas

**Fig. 2.** The distribution of attendance**Table 2.** Descriptive statistics for attendance rate

Variable	Operas	Ballets	Total
<i>Attendance rate</i>	0.72	0.96	0.80
Attendance (area 1)	0.83	0.92	0.85
Attendance (area 2)	0.87	0.96	0.89
Attendance (area 3)	0.85	0.96	0.89
Attendance (area 4)	0.86	0.97	0.90
Attendance (area 5)	0.76	0.98	0.84
Attendance (area 6)	0.68	0.98	0.80
Attendance (area 7)	0.52	0.97	0.70
Attendance (area 8)	0.46	0.95	0.65
Attendance (area 9)	0.64	0.87	0.72

compared to ballets. The quality of seat in terms of the view and sound quality should be also taken into account in a model of demand with an attention to potentially different seats quality estimate for various production types. It also may result in different estimates of price elasticity over types of production and seats since willingness-to-pay for a particular seat associated with its quality may vary over operas and ballets.

3 Methodology

Since the theatre attendees have various goals for visiting the theatre, they have different willingness-to-pay for performance and play characteristics and seats in the house. Self-segmenting of consumers among performances and seats

determines a heterogeneity in theatre demand. ML methods for prediction of a theatre attendance need to be focused on a model selection based on the partition of data space according to consumer and product segments. Tree-based methods include such property and are used in the paper. However, existed methods such as CART (Breiman *et al.* 1984) and its more recent extensions deal only with classification or continuous data prediction problems but with a problem of censored data prediction. In this paper we develop a CART-based algorithm addressed to a prediction of censored data. Then we start with a description of known tree methods and follow with 4-step algorithm that accounts for data censorship.

A regression tree is a collection of rules that determine the value of a function. Tree-based methods partition the characteristic space into a series of hypercubes and fit effects to each partition depend on the X . Trees are characterized by a hierarchical series of nodes, with a decision rule associated at each node. Following (Bajari *et al.* 2015), define a pair of half-planes:

$$R_1(j; s) = X|X_j \leq s \quad (1)$$

$$R_2(j; s) = X|X_j > s$$

where j indexing a splitting variable and s is a split point. Starting with the base node at the top of the tree, the rule for that node is formed by the following optimization problem:

$$\min_{j,s} [\min_{\theta_1} \sum_{i:x_i \in R_1(j,s)} L(y_i - f(x_i|\theta_1)) + \min_{\theta_2} \sum_{i:x_i \in R_2(j,s)} L(y_i - f(x_i|\theta_2))] \quad (2)$$

The inner optimization is solved by setting θ optimal according to prespecified loss function L and regression function f . For ordinary regression tree L is a squared function and f is a linear function of x with parameters θ . We employ a tree of median regressions setting L as an absolute deviation function to control for influential observations. In a classification problem, the classification tree is built based on binary choice (probit) function f with a loss L associated with errors in classification. We use ordinary classification accuracy measure, a number of misclassified observations, since relative importance of type I and II errors is not defined.

The outer optimization problem is a problem of finding an optimal splitting point s for each possible splitting variable and then choosing a variable x to split by. Once the splitting variable and point are found, the same procedure is then performed on each resulting partitioning, resulting in a partition of characteristics space.

In the limit, each value of $x \in X$ is assigned to value of $y = f(x)$, which is a perfect reconstruction of the underlying function f for in-sample prediction. In practice, we are interested in out-of-sample prediction. Therefore, the tree is expanded until a value of loss function for out-of-sample data falls. Often, the tree is grown until a specific number of splits or a minimal number of observations in subsamples is achieved.

The literature has proposed several variations on the regression tree estimator to obtain “honest” prediction and predicted values robust to influential observations. One is bagging (Breiman 1996), which uses resampling and model combination to obtain a predictor. The idea is to sample the data with replacement B times, train a regression tree on each resampled set of data, and then predict the outcome at each x through a simple average of the predictions under each of the B trees.

Since we have a problem of censored data prediction, we construct an algorithm for prediction of both discrete and continuous components of dependent variable x and a combination of these predictions. An algorithm has following steps:

1. Construct dummy for observation censorship $d := I\{y = 1\}$.
2. Classify observations into censored and uncensored ($\hat{d} \in \{0; 1\}$) based on bagging prediction from classification (probit) trees and predict $\hat{p} = E[d|X]$;
3. Predict y using median regression tree trained on classified as uncensored ($\hat{d} = 0$) data with \hat{p} as predictor:

$$\hat{y} = \min\{Q_{y|X,\hat{p}}(0.5); 1\}$$

4. Combine the predictions of discrete (\hat{d}) and continuous (\hat{y}) components:

$$\hat{\hat{y}} = \begin{cases} 1, \hat{d} = 1 \\ \hat{y}, \hat{d} = 0 \end{cases}$$

4 Results

Firstly, we compare the predictive accuracy of the proposed estimator compared with parametric and nonparametric ones to predict seating area attendance rate. We perform 4 parametric estimators and construct bagging predictions similar to the proposed tree-based algorithm. Two parametric estimators (OLS and quantile regression) do not account for censorship and heterogeneity while two more (Tobit model and censored quantile regression) account for censorship only. Tree of median regressions accounts for heterogeneity but not for censorship. Our estimator (tree of censored quantile regressions) outperforms all estimators in terms of prediction error and better explains variance compared to parametric estimators. Results presented in Table 3 show that given the data on theatre demand it is necessary to account for demand censorship and heterogeneity.

Secondly, we analyze the variables the importance for trees grown calculating the share of partitions by a certain variable among all partitions in estimated trees. Importance of variables for data partition shows the main sources of heterogeneity of effects which matters for demand prediction. We separately calculate importance for growing trees for prediction of discrete (\hat{d}) and continuous (\hat{y}) and parts of the data. Results for variables importance are presented in Table 4.

Results vary for prediction of \hat{d} and \hat{y} since they are estimated on different subsamples of the data. A model for \hat{d} is calibrated on the whole sample while a

Table 3. Prediction power of estimators

y	Mean	SD	R^2	RMSE	Min	Max
	0.803	0.263			0.009	1
Model for \hat{y}						
CQR Tree	0.813	0.201	0.588	0.052	0.089	1
CQR	0.823	0.171	0.422	0.083	0.209	1
Tobit	0.823	0.183	0.488	0.080	0.188	1
QR Tree	0.804	0.206	0.618	0.059	0.043	1
QR	0.842	0.121	0.212	0.098	0.459	1
OLS	0.793	0.159	0.370	0.110	0.310	1
Number of observations	2682					
Number of predictors	36					
Number of replications	200					

Table 4. Variables importance

Variable	Share of splits		
	\hat{d}	\hat{y}	Total
Seating area	0.142	0.032	0.105
Premiere	0.080	0.105	0.088
Laureat of GM	0.244	0.050	0.180
Ballet	0.043	0.077	0.055
Rated opera	0.048	0.048	0.048
Rated ballet	0.095	0.153	0.114
Russian composer	0.085	0.052	0.074
Foreign language	0.043	0.087	0.058
Band director: Platonov	0.083	0.164	0.110
Band director: Currentzis	0.007	0.009	0.008
12+	0.082	0.098	0.087
16+	0.013	0.032	0.019
Evening	0.010	0.025	0.015
Friday	0.000	0.007	0.002
Saturday	0.002	0.025	0.010
Sunday	0.023	0.036	0.027
Number of trees	200	200	200
Number of splits	1204	1253	2457
Mean number of splits in a tree	6.0	6.1	6.0

model for \hat{y} is calibrated only on the expectedly uncensored observations. Results show that the main sources of observations heterogeneity are those related to prestige of tickets (seating area, dummy for premiere play and dummy for play of “Golden Mask” nominee), content of performance (type of performance, world rating of the production, nationality of composer, language of opera singing and recommended age) and band director while time and day of performance have only small explanation of heterogeneity. Among all of the heterogeneous effects on demand, we are interested in studying the heterogeneity in price elasticity of demand. To calculate the price elasticity we construct two predictions of attendance, the first one is a prediction with current prices and the second one is a prediction with prices all increased by 10% from the current level. We find the total elasticity range from -0.3 to 0 , that corresponds to a weak elasticity of demand (See Fig. 3). Zero elasticity for the substantial share of observations indicates that for fully occupied seating areas the potential demand is significantly higher than capacity. The increase of the price by 10% will decrease the potential demand but not below the capacity level. Then the observed demand will remain on the same level.

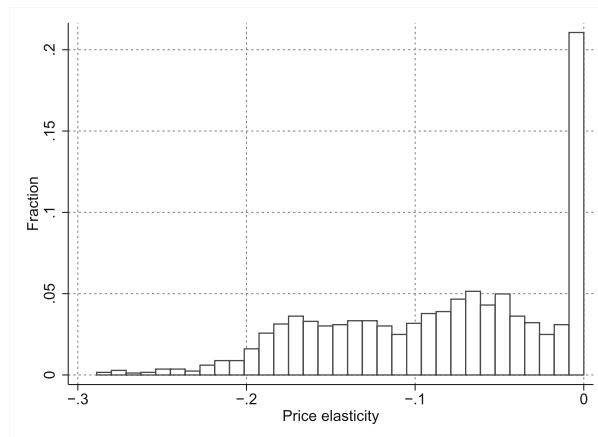


Table 5. Price elasticity results by samples

Variable	Operas	Ballets	Total
All seating areas	-0.107	-0.043	-0.086
Seating area 1	-0.100	-0.059	-0.086
Seating area 2	-0.097	-0.030	-0.074
Seating area 3	-0.098	-0.028	-0.074
Seating area 4	-0.100	-0.032	-0.076
Seating area 5	-0.109	-0.049	-0.088
Seating area 6	-0.116	-0.038	-0.089
Seating area 7	-0.123	-0.048	-0.101
Seating area 8	-0.111	-0.050	-0.092
Seating area 9	-0.113	-0.056	-0.096
Rated	-0.096	-0.033	
Non-rated	-0.112	-0.054	
Russian composer	-0.130	-0.038	-0.103
Foreign composer	-0.073	-0.048	-0.063
Russian language	-0.086		
Foreign language	-0.062		
Band director: Others	-0.111	-0.044	-0.087
Band director: Currentzis	-0.085	-0.016	-0.075
Other days	-0.087	-0.017	
Friday	-0.038		
Saturday		-0.004	

5 Conclusion

In this research we analyze the demand for performing arts on the ticket sales data obtained from Perm Opera and Ballet Theatre. Data contain information on the attendance of seating areas for 298 performances played in 2011–2015. Since the observed demand is limited by the capacity of the house and the third of seating areas are fully occupied, one needs to account for demand censorship. The presence of consumer segments with different purposes of going to the theatre and willingness-to-pay for performance and ticket characteristics causes a heterogeneity in theatre demand.

We propose an estimator for prediction of demand that accounts for both demand censorship and preferences heterogeneity. The estimator is based on the idea of classification and regression trees (CART) and bagging prediction aggregation. We extend CART for the problem censored dependent variable prediction. The algorithm consists of three steps: (1) Bagging prediction of dummy whether the dependent variable is on the censoring bound using classification trees; (2) Bagging prediction of the dependent variable for observations

classified as uncensored using median regression trees; (3) Trimming of second-step prediction and its combination with the first-step prediction of censored observations.

We find a better performance of our algorithm in terms of predictive power compared with parametric methods (OLS and median regression), parametric methods (Tobit model and censored quantile regression) which account for data censorship but not for heterogeneity and the method (tree of quantile regressions) that accounts for heterogeneity only. We study the importance of variables for the explanation of demand heterogeneity. The most frequent variables for splitting the sample on subsamples with different effects on demand are the type of production (ballet or opera), a seating area, the nationality of composer (Russian or foreign) and a band director. We estimate the distribution of price effect comparing the predictions of attendance with current prices and prices increased by 10%. The price elasticity varies from -0.30 to 0 with a median equal to -0.07 that indicates on weakly elastic demand. We find that price elasticity of demand varies substantially with less elastic demand for ballets, Russian ballets among ballets and foreign operas among operas, and seats in the center of stalls.

Acknowledgement. The publication was prepared within the framework of the Academic Fund Program at the National Research University Higher School of Economics (HSE) in 2018-2019 (grant № 18-01-0025 and by the Russian Academic Excellence Project “5-100”).

References

- Bajari, P., Nekipelov, D., Ryan, S.P., Yang, M.: Machine learning methods for demand estimation. *Am. Econ. Rev.* **105**(5), 481–485 (2015)
- Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. CRC Press, Boca Raton (1984)
- Breiman, L.: Bagging predictors. *Mach. Learn.* **24**(2), 123–140 (1996)
- Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
- Chernozhukov, V., Hong, H.: Three-step censored quantile regression and extramarital affairs. *J. Am. Statist. Assoc.* **97**(459), 872–882 (2002)
- Chernozhukov, V., Fernandez-Val, I., Kowalski, A.E.: Quantile regression with censoring and endogeneity. *J. Econometrics* **186**(1), 201–221 (2015)
- Corning, J., Levy, A.: Demand for live theater with market segmentation and seasonality. *J. Cult. Econ.* **26**(3), 217–235 (2002)
- Das, M., Newey, W.K., Vella, F.: Nonparametric estimation of sample selection models. *Rev. Econ. Stud.* **70**(1), 33–58 (2003)
- Heckman, J.: Sample selection bias as a specification error. *Econometrica* **47**, 153–161 (1977)
- Matzkin, R.L.: Identification in nonparametric limited dependent variable models with simultaneity and unobserved heterogeneity. *J. Econometrics* **166**(1), 106–115 (2012)
- Seaman, B.A.: Empirical studies of demand for the performing arts. In: *Handbook of the Economics of Art and Culture*, vol. 1, pp. 415–472 (2006)
- Tobin, J.: Estimation of relationships for limited dependent variables. *Econometrica J. Econometric Soc.* **26**, 24–36 (1958)

Original Loop-Closure Detection Algorithm for Monocular vSLAM

Andrey Bokovoy^{1,3(✉)} and Konstantin Yakovlev^{2,3}

¹ Peoples' Friendship University of Russia (RUDN University), Moscow, Russia
1042160097@rudn.university

² Higher School of Economics, Moscow, Russia
kyakovlev@hse.ru

³ Institute for Systems Analysis of Federal Research Centre
“Computer Science and Control” of Russian Academy of Sciences, Moscow, Russia
{bokovoy,yakovlev}@isa.ru

Abstract. Vision-based simultaneous localization and mapping (vSLAM) is a well-established problem in mobile robotics and monocular vSLAM is one of the most challenging variations of that problem nowadays. In this work we study one of the core post-processing optimization mechanisms in vSLAM, e.g. loop-closure detection. We analyze the existing methods and propose original algorithm for loop-closure detection, which is suitable for dense, semi-dense and feature-based vSLAM methods. We evaluate the algorithm experimentally and show that it contribute to more accurate mapping while speeding up the monocular vSLAM pipeline to the extent the latter can be used in real-time for controlling small multi-rotor vehicle (drone).

Keywords: Loop-closure · Vision-based localization and mapping
Unmanned aerial vehicle · SLAM · vSLAM

1 Introduction

Vision-based simultaneous localization and mapping (vSLAM) is one of the most challenging problems in computer vision and robotics. SLAM methods, that rely only on the information gained from minimum set of miniature passive sensors (monocular or stereo camera, inertial measurement unit), lie at the core of navigation capabilities of various mobile robots. Especially, they are of great value for compact unmanned aerial vehicles (which can not be equipped by the heavy, powerful sensors by default).

Recently a notable progress in the field of UAV vSLAM methods was made, see [1, 2], for example. However, there's still a large set of real-world problems and scenarios that can not be successfully tackled by the existing vision-based SLAM algorithms. The main reasons for that are the following.

First is the image processing time. Modern embedded computers that can be installed on compact UAVs are not that powerful to execute typical vSLAM

pipelines in real time. Using external sources for remote computations is not always the solution since it lowers the mobility (robotic system is forced to continuously exchange huge amount of information with remote control station, using wire or wireless channel) and prevents robotic system from being fully autonomous.

Second is poor image quality [3]. Small cameras typically mounted on compact UAVs are highly affected by the environment's conditions (light, weather etc.) and often produce video stream containing numerous jitters, noises and other artifacts. Thus one needs to apply different filtering techniques to pre-process the video stream and thus to improve the efficiency of vSLAM methods.

On top of that, all vSLAM methods are prone to accumulating error [4] and that negatively affects the accuracy of constructed map and trajectory. One way to correct this error, and thus to increase the overall performance, is to handle, i.e. detect, *loop-closures* - see Fig. 1. More precisely one needs to detect that the current image comes from an already perceived scene and, in case it's true, correct the map and the trajectory.

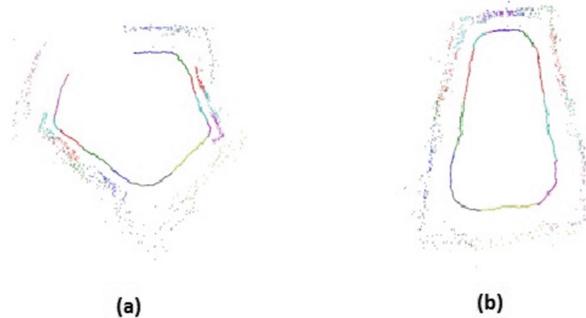


Fig. 1. Solving SLAM problem with and without use of the loop-closure algorithm. (a) A raw map obtained with monocular vision-based SLAM method. The inner curve represents the trajectory of mobile robotic system. The outer points represent the map. (b) Trajectory and map optimized with the loop-closure algorithm

In this paper, we focus on improving the accuracy and performance of loop-closure detection algorithms. The ultimate goal is to keep the algorithm as robust and fast as possible along with making it compatible with dense, semi-dense and feature-based vSLAM methods. We introduce two enhancement steps (within the loop-closure detection algorithm) that contribute towards reaching this goal.

The latter of the paper is organized as follows. In Sect. 2, we present a brief overview of existing methods. Section 3 introduces our implementation of loop-closure algorithm. The experimental results, showing the accuracy and performance of implemented algorithm, are given in Sect. 4. Section 5 concludes.

2 Loop-Closure Methods

Accumulating error is one of the main bottlenecks of almost all known monocular vSLAM methods and algorithms. Even state-of-the-art algorithms suffer from this [5]. At the same time, results of numerous feasibility studies show, that detecting loop-closures can drastically improve the overall performance of monocular vSLAM. No wonder many of the vSLAM methods have loop-closure detection procedures built-in [6–8]. There exist also standalone loop-closure detectors [9,10] that may be plugged in to some of the vSLAM methods.

The earlier work [11–13] mostly rely on the so-called global loop detection, when the current image was compared against all previous visual data. This approach is quite reliable, but comes at the cost of high computation load and memory usage as one needs to keep all the information (such as keypoints, intense areas, depth map etc.) for every image processed during algorithm runtime. This leads to poor scaling for large environment localization and mapping. The recent approaches [14–18] use different constraints (i.e. using keyframes for keypoint matching) to optimize the time required for loop detection and map correction, but their usage is usually limited to specific vSLAM method.

In general loop-closure detection algorithms can be classified into three groups [19]: **map-to-map**, **image-to-map** and **image-to-image**:

- **map-to-map** loop closure is done by splitting the global map into sub-maps and finding correspondences between them [20].
- **image-to-map** performs the search of the matches between image and a map and recovers the system’s position, relative to the map [21].
- **image-to-image** finds a correspondences between images, usually based on vocabulary of image features [22].

Map-to-map approach is very intense performance-wise, since it deals with large amount of information on each iteration while comparing sub-maps. As the result it scales poorly to large environments. Image-to-map approach is fast and accurate, but in practice it is very memory intensive because one needs to store both point-cloud map and all the image features. The image-to-image loop-closure scales well to large environments, and can be computed fast with feature based approaches, but highly relies on a vocabulary. Thus one can infer that a combination of different approaches is desirable to reach higher performance while keeping the accuracy and the robustness at the high level. In this work we propose a solution that contributes towards this goal.

Proposed loop-closure method aims to combine image-to-image and image-to-map approach to achieve scalability, robustness and accuracy of both approaches, while keeping moderate runtime and low memory usage. Besides the proposed method is compatible with a large number of existing vSLAM methods, including feature-based, semi-dense and dense vision-based SLAM methods (for monocular, stereo and RGB-D cameras) and can be seen as a general enhancement approach to loop-closure detection.

3 Proposed Method

In a nutshell all loop-closure algorithms generally consist of the two steps: (1) loop detection, (2) global optimization. Loop detection aims at establishing that the particular image is part of the scene, that has already been captured by previous image sequences. The simple interpretation is that this may be a sign, that the robotic system has reached the place that had already been visited before. The global optimization is performed after the loop is detected. This step corrects the accumulated run-time error for both the map and the trajectory (in a background). The illustration of the loop-detection process is depicted on Fig. 2.

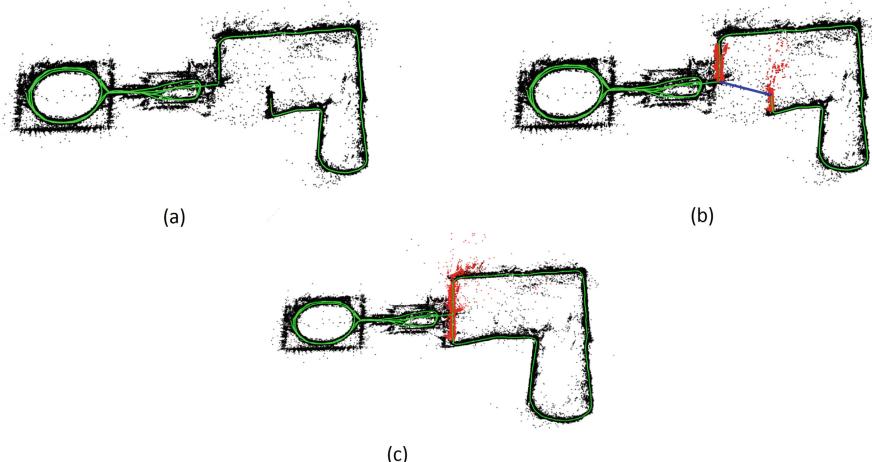


Fig. 2. Main steps of the loop-closure detection. (a) Given (current) setting. Green curve represents the trajectory of the robotic system, black points represent the mapped features. (b) The loop-detection step. Red points represent the matched features, e.g. the ones that are present on the current image (perceived in current position) and previously observed images. (c) Map and trajectory of the robot after loop-closure detection. (Color figure online)

Since the robotic system's motion consists of continuous rotations and translations, we assume that the trajectory is continuous as well (unless vSLAM method's tracking is lost), so loop-detection algorithm usually checks for trajectory loops once per N images for performance optimization purposes. In cases, when tracking is lost, detection may be needed to recover the state and position of robotic system and rebuild the map.

We suggest 2 enhancement procedures to be performed while detecting the loop. They both aim at lowering down the number of features to be compared thus speeding up the algorithm. The enhancements include the image detection optimization and imposing geometric constraints. For fast and accurate image

matching we found that storing a particular amount of informative keypoints (instead of all keypoints) for each image allows us to keep the image matching accuracy. Also, the keypoints search area can be reduced to the only mapped points. The image comparison search area can be reduced by the geometric constrained, that is based on current camera position. We choose only images from the field, that may be observed from the camera in current position. High-level pseudocode of the loop-detection algorithm with the aforementioned procedures built-in is shown as Algorithm 1.

Algorithm 1. The proposed loop-closure detection algorithm.

1. Get an image from video flow
 2. Extract keypoints and get their descriptors from corresponded mapped points on image
 3. Get and store \mathbf{K} informative features from image
 4. **if** The trajectory loop is in camera search area
 5. Match corresponding images in search area with current image
 6. **if** the correspondence found
 7. Perform the map optimization
 8. **endif**
 9. **endif**
-

First procedure (lines 2-3) affects the feature extraction area of image. We rely only on dense and semi-dense vision-based SLAMs that were used for depth map computation and mapping purposes, e.g. points with high gradient of intensity. Thus, we reduce the extraction area by using only high gradient pixels, that were previously chosen to reconstruct the 3D space from 2D image (line 2). This allows us to avoid the image areas that are not going to be mapped anyway and provides an opportunity to reduce feature extraction process time. We limit the keypoints amount per any image in video flow to K (line 3). These K keypoints with their descriptors are stored during loop-closure algorithm run-time since the number of keypoints per frame is relatively small (see Sect. 4).

Second procedure (line 4) is the loop detection search area limitation. This allows to identify the patch on the whole trajectory that, with high probability, has a loop-closure point in it (i.e. the place, where the robot has already been). Assuming the robotic system's motion is mostly horizontal, we project the motion vector and continue it with a straight line. Then we draw a perpendicular to this line. If the perpendicular intersects the built trajectory, then we draw a α degree line between normal and the projected motion line. The closest position (with corresponding image) to the point of intersection is going to be a start point for loop detection algorithm with the whole loop detection area constrained by two points - the intersection of normal and motion line with trajectory.

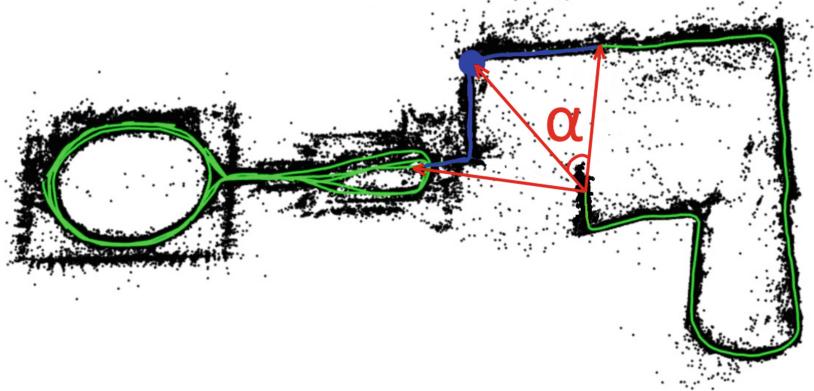


Fig. 3. Loop detection search area. Red vectors shows the bounds for image comparison. Blue dot represents the start point for loop detection algorithm. And the trajectory picked out with blue color is our search area. (Color figure online)

If motion line has no intersection point, then a search starts from the initial position of vision-based SLAM algorithm. The illustration of suggested method is demonstrated in Fig. 3.

More formally one can put it as follows. Assuming, that raw localization and mapping (without optimization) for each moment of time t is done by vSLAM method. Thus, for a given moment of time t , we have a point cloud $M = \{m_i\}, i \in N$, that represents the map, sequence of images $I = \{I_1, I_2, \dots, I_t\}$. For each image I_t we have corresponding observation $z_t \in M$ and position vector x_t .

As a part of loop-closure detection algorithm, we project each position x_t on xy plain $x'_t = Px_t$. For each $x'_k, k < t - 1, k \in N$ we check if vector $\vec{w} = \overrightarrow{x'_{t-1}sx'_t}, s > 0$ has intersection point p with any of vectors $\vec{v} = \overrightarrow{x'_{k-1}x'_k}$ and vector $\overrightarrow{x'_1(-lx'_2)}, l > 0$. We assume the position x'_k closest to intersection point p to be the starting point for image matching. As an end point for image matching, we take the intersection point p' of perpendicular h to vector $\overrightarrow{x_t, x_{t-1}}$. As the result, the current image matching with corresponded images from positions between points p and p' .

3.1 Implementation

As the main image identifier for loop detection we've chosen ORB detector [23] as one of the most fast, robust and efficient feature detector. For each image we extract at least K ORB features and store their oriented and rotated BRIEF [24] descriptors, that have high element sum, with associated images. The requirement of having element sum in BRIEF descriptors comes from their interpretation. Higher values mean higher intensity gradient at this points, that provides

more robust feature matching. That means that such a keypoints are informative and can be stored for further image matching.

As a part of map and trajectory global optimization, we use one of the most popular and effective graph optimization framework g²o [25]. That allows us to keep a high accuracy while optimizing map and trajectory in comparison to other modern vSLAM methods.

4 Experimental Results

For performance and accuracy testing purposes of the developed method we use a Robot Operating System (ROS) [26], that provides a powerful tools for robotic algorithms researches in general and in vision-based SLAM testing in particular. The open-source realizations of ORB-SLAM and LSD-SLAM were taken as ones of the most popular feature-based and semi-dense SLAMs respectively.

Table 1. Loop detection success table

Dataset	Method	ORB Features										
		8	9	10	11	12	13	14	15	16	17	18
Sequence 13	ORB-SLAM	-	-	-	-	+	+	+	+	+	+	+
	LSD-SLAM	-	-	-	-	-	-	+	+	+	+	+
Sequence 14	ORB-SLAM	-	-	+	+	+	+	+	+	+	+	+
	LSD-SLAM	-	-	-	+	+	+	+	+	+	+	+
Sequence 15	ORB-SLAM	-	+	+	+	+	+	+	+	+	+	+
	LSD-SLAM	-	+	+	+	+	+	+	+	+	+	+
Machine	ORB-SLAM	-	-	-	-	-	-	+	+	+	+	+
	LSD-SLAM	-	-	-	-	-	-	-	+	+	+	+
Foodcourt	ORB-SLAM	-	-	-	-	-	-	+	+	+	+	+
	LSD-SLAM	-	-	-	-	-	-	+	+	+	+	+

The introduced method is used with raw point cloud output of this methods. The experiment was made using LSD-SLAM Dataset¹, KITTI vision benchmark suit [27, 28]² and Malaga Dataset [29]³, which video fragments was divided into subsequences (distinguishing fragments with loops) to make the experimental research more relevant.

We took the Sequences 13, 14 and 15 from KITTI dataset and Machine and Foodcourt Sequences from LSD-SLAM dataset, because that sequences contain trajectories with loop-closures. KITTI dataset includes ground truth, that allows

¹ <http://vision.in.tum.de/research/vslam/lslsdlam>.

² http://www.cvlibs.net/datasets/kitti/eval_odometry.php.

³ <http://www.mrpt.org/MalagaUrbanDataset>.

us to compare the optimized trajectory with real one. For LSD-SLAM datasets we only test the performance of our algorithm and the accuracy in comparison with trajectories, built by LSD-SLAM and ORB-SLAM. For Malaga Dataset we took the whole 6th, 7th and 8th sequence, since they present single loop, and sequences 10 and 13 (which contain multiple loops) where divided into 7 and 3 single loop subsequences respectively. Thus, we used 13 sequences from Malaga Dataset.

The first experiment was made to test the minimum required ORB features (K value) for loop detection algorithm to function successfully (e.g. with 100% success rate). The Fig. 4 shows the results of such an experiment. The Table 1 shows the if the loop was successfully detected depending on number of ORB features (R) used for image matching (Figs. 5 and 7).

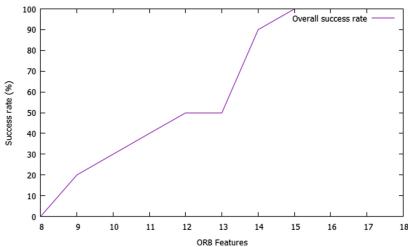


Fig. 4. Overall success rate of loop-closure detection algorithm with different amount of keypoints.

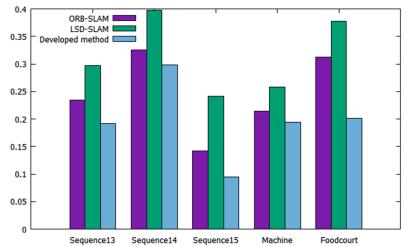


Fig. 5. The histogram shows run time (in seconds) for loop-closure algorithms used in ORB-SLAM and LSD-SLAM in comparison with our algorithm.

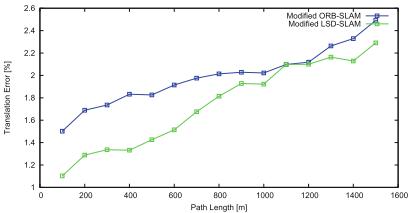


Fig. 6. Average translation error over 1.5 km distance for ORB-SLAM and LSD-SLAM with proposed loop-closure detection method.

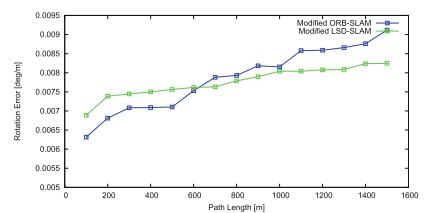


Fig. 7. Average rotation error over 1.5 km distance for ORB-SLAM and LSD-SLAM with proposed loop-closure detection method.

As was already mentioned in Sect. 2, we need to store at least K features to successfully match the images if the loop-closure occurred. The experimental data shows that $K = 15$ is a minimum value for loop to be detected. The presented result also allows us to dramatically reduce the memory usage, since we don't have to store hundreds of BRIEF descriptors, and increase the overall

performance by the average of 7-10% in comparison with LSD-SLAM's and ORB-SLAM's loop-closure algorithms as shown in Fig. 8.

For KITTI and Malaga sequences, the trajectory ground truth is presented, so we tested our algorithm using the available data. Figure 6 shows the ground truth trajectory and the trajectory optimized with our method. The overall error values vary from 1.5% to 2.5% that is comparable with LSD-SLAM's and ORB-SLAM's loop-closure precision.

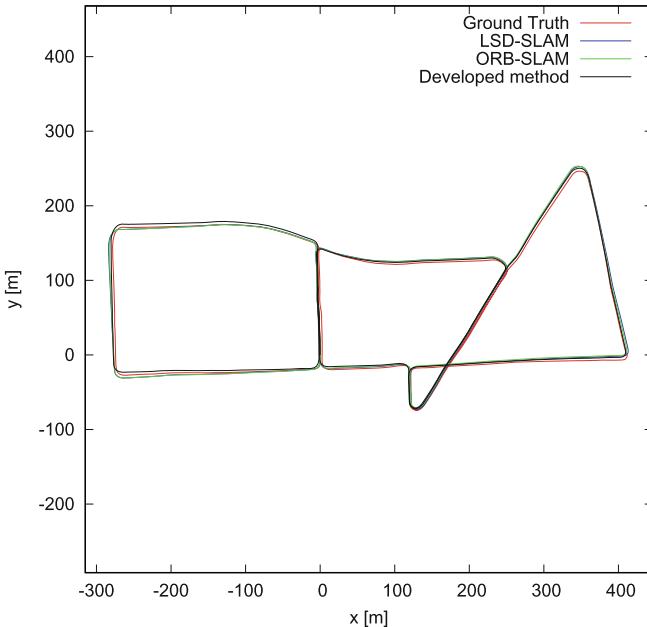


Fig. 8. Ground truth comparison between LSD-SLAM, ORB-SLAM and developed method for sequence 13.

The overall precision depends on trajectory's length and geometry. We found that longer trajectories with multiple loops give more accurate trajectory optimization for our method, while being more time consuming.

5 Conclusion

We have developed the original loop-closure method, that can be used for dense, semi-dense and feature-based vSLAM methods. The introduced optimization techniques showed, that the combination of image-to-image approach for loop detection and image-to-map approach for global optimization keeps an accurate trajectory error correction (around 1.5-2.5% translation error) while decreasing process time by 7-10%.

We found, that introduced method works in large outdoor environment without major issues. The experimental results showed, that described method can be used for mini unmanned aerial vehicle autonomous navigation tasks, even onboard.

Acknowledgment. This research was supported by Russian Foundation for Basic Research. Grant 15-07-07483.

References

1. Fu, C., Olivares-Mendez, M.A., Suarez-Fernandez, R., Campoy, P.: Monocular visual-inertial slam-based collision avoidance strategy for fail-safe UAV using fuzzy logic controllers. *J. Intell. Robot. Syst.* **73**(1–4), 513–533 (2014)
2. Weiss, S., Scaramuzza, D., Siegwart, R.: Monocular-slam-based navigation for autonomous micro helicopters in GPS-denied environments. *J. Field Robot.* **28**(6), 854–874 (2011)
3. Handa, A., Whelan, T., McDonald, J., Davison, A.J.: A benchmark for RGB-D visual odometry, 3D reconstruction and slam. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 1524–1531. IEEE (2014)
4. Strasdat, H., Montiel, J.M., Davison, A.J.: Visual slam: why filter? *Image Vis. Comput.* **30**(2), 65–77 (2012)
5. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. In: [arXiv:1607.02565](https://arxiv.org/abs/1607.02565), July 2016
6. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8690, pp. 834–849. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10605-2_54
7. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular slam system. *IEEE Trans. Robot.* **31**(5), 1147–1163 (2015)
8. Mur-Artal, R., Tardós, J.D.: Visual-inertial monocular slam with map reuse. *IEEE Robot. Autom. Lett.* **2**(2), 796–803 (2017)
9. Angeli, A., Filliat, D., Doncieux, S., Meyer, J.A.: Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Trans. Robot.* **24**(5), 1027–1037 (2008)
10. Strasdat, H., Davison, A.J., Montiel, J.M., Konolige, K.: Double window optimisation for constant time visual slam. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2352–2359. IEEE (2011)
11. Botterill, T., Mills, S., Green, R.: Bag-of-words-driven, single-camera simultaneous localization and mapping. *J. Field Robot.* **28**(2), 204–226 (2011)
12. Konolige, K., Bowman, J., Chen, J., Mihelich, P., Calonder, M., Lepetit, V., Fua, P.: View-based maps. *The Int. J. Robot. Res.* **29**(8), 941–957 (2010)
13. Cummins, M., Newman, P.: Probabilistic appearance based navigation and loop closing. In: 2007 IEEE International Conference on Robotics and Automation, pp. 2042–2048. IEEE (2007)
14. Mur-Artal, R., Tardós, J.D.: Probabilistic semi-dense mapping from highly accurate feature-based monocular slam. In: Robotics: Science and Systems (2015)
15. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: RGB-D mapping: using depth cameras for dense 3D modeling of indoor environments. In: Experimental Robotics, pp. 477–491. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-28572-1_33

16. Vokhmintsev, A., Timchenko, M., Yakovlev, K.: Simultaneous localization and mapping in unknown environment using dynamic matching of images and registration of point clouds. In: International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), pp. 1–6. IEEE (2016)
17. Buyval, A., Gavrilenkova, M.: Vision-based pose estimation for indoor navigation of unmanned micro aerial vehicle based on the 3D model of environment. In: 2015 International Conference on Mechanical Engineering, Automation and Control Systems (MEACS), pp. 1–4. IEEE (2015)
18. Afanasyev, I., Sagitov, A., Magid, E.: ROS-based SLAM for a gazebo-simulated mobile robot in image-based 3D model of indoor environment. In: Battiato, S., Blanc-Talon, J., Gallo, G., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2015. LNCS, vol. 9386, pp. 273–283. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25903-1_24
19. Eade, E., Drummond, T.: Unified loop closing and recovery for real time monocular slam. In: BMVC, vol. 13, p. 136 (2008)
20. Clemente, L.A., Davison, A.J., Reid, I.D., Neira, J., Tardós, J.D.: Mapping large loops with a single hand-held camera
21. Williams, B., Cummins, M., Neira, J., Newman, P., Reid, I., Tardós, J.: An image-to-map loop closing method for monocular slam. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2008, pp. 2053–2059. IEEE (2008)
22. Cummins, M., Newman, P.: Fab-map: probabilistic localization and mapping in the space of appearance. Int. J. Robot. Res. **27**(6), 647–665 (2008)
23. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to sift or surf. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2564–2571. IEEE (2011)
24. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: binary robust independent elementary features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 778–792. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_56
25. Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: g2o: a general framework for graph optimization. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 3607–3613. IEEE (2011)
26. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source robot operating system. In: ICRA Workshop On Open Source Software, vol. 3, p. 5, Kobe (2009)
27. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the kitti dataset. Int. J. Robot. Res. **32**(11), 1231–1237 (2013)
28. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition CVPR (2012)
29. Blanco-Claraco, J.L., Moreno-Dueñas, F.Á., González-Jiménez, J.: The málaga urban dataset: high-rate stereo and lidar in a realistic urban scenario. Int. J. Robot. Res. **33**(2), 207–214 (2014)

Analysis of Images and Video

Organizing Multimedia Data in Video Surveillance Systems Based on Face Verification with Convolutional Neural Networks

Anastasiia D. Sokolova^(✉), Angelina S. Kharchevnikova,
and Andrey V. Savchenko

National Research University Higher School of Economics,
Nizhny Novgorod, Russian Federation
adsokolova96@mail.ru

Abstract. In this paper we propose the two-stage approach of organizing information in video surveillance systems. At first, the faces are detected in each frame and a video stream is split into sequences of frames with face region of one person. Secondly, these sequences (tracks) that contain identical faces are grouped using face verification algorithms and hierarchical agglomerative clustering. Gender and age are estimated for each cluster (person) in order to facilitate the usage of the organized video collection. The particular attention is focused on the aggregation of features extracted from each frame with the deep convolutional neural networks. The experimental results of the proposed approach using YTF and IJB-A datasets demonstrated that the most accurate and fast solution is achieved for matching of normalized average of feature vectors of all frames in a track.

Keywords: Organizing video data · Video surveillance system · Deep convolutional neural networks · Clustering · Face verification

1 Introduction

Nowadays, due to the growth of the multimedia data volume, the task of forming an automatic approach to the ordering of digital information is attracting increasing attention [1]. The various photo organizing systems allows the user to speed up the search for the required frame, and also to increase the efficiency of work with the media library. Such modern solutions include services like Apple iPhoto, Google Photos, etc., which are designed to store, organize and display media data. However, multimedia data organization systems are required not only for a particular user who has an archive of photographs, but also for the field of public safety, where video surveillance technologies are used for monitoring purposes [2]. Consequently, there is a challenge of ordering the visitors, whose faces are observed in a surveillance system. To solve the problem the clustering of video tracks that contains the same person can be performed using

the known face verification methods [3, 4] based on deep convolutional neural networks (CNNs) [5–8]. Unlike the traditional technologies of ordering digital information, video surveillance systems are characterized by a large amount of data, because hundred frames can be obtained in dynamics in a few seconds [4, 9]. Therefore the goal of our research is to improve the verification efficiency by the combination for features extracted from individual frames. The rest of the paper is organized as follows: in Sect. 2, we formulate the proposed approach of organizing multimedia data in video surveillance system. In Sect. 3, we present the experimental results in unconstrained face verification. Concluding comments are given in Sect. 4.

2 Automatic Organization of Video Data

The task of this paper is to split the given video sequence of T frames into subsequences with observations of one person, and then unite different subsequences containing the same person. At first, the facial regions are detected [9] using, e.g., the Viola-Jones method. For simplicity, we assume that each frame in the given video consists of images (frames) of exactly one face. Next, an appropriate tracker algorithm [9, 10] divides the input sequence into $M < T$ disjoint subsequences (tracks) $\{X(m)\}$, $m = 1, 2, \dots, M$, where the m -th frame is characterized by its borders $(t_1(m), t_2(m))$, where the m -th track contains $\Delta t(m) = t_2(m) - t_1(m) + 1$ frames. Finally, we search for similar tracks using, e.g., hierarchical agglomerative clustering methods [11]: similar objects are sequentially grouped together. In order to implement any clustering method, a dissimilarity measure between video tracks should be defined. Let us extract appropriate facial features from every frame.

Nowadays feature extraction is implemented using the deep CNNs trained with an external large dataset, e.g., Casia WebFaces or MS-Celeb-1M [6, 12]. The outputs of the CNN's last (bottleneck) layer for the t -th frame are stored in the D -dimensional feature vector $x(t)$. These bottleneck features are usually matched with the Euclidean distance $(x(t_1), x(t_2))$ [12]. It is possible to define the dissimilarity of tracks $X(m_1)$ and $X(m_2)$ as a summary statistic of distances between individual frames. In our experiments the highest accuracy was achieved with the average distance:

$$\rho(X(m_1), X(m_2)) = \frac{1}{\Delta t(m_1) \Delta t(m_2)} \sum_{t=t_1(m_1)}^{t_2(m_1)} \sum_{t'=t_1(m_2)}^{t_2(m_2)} \rho(x(t), x(t')). \quad (1)$$

However, the run-time complexity of such distance is high due to the pair-wise matching of all frames in these tracks causing the computation of $\Delta t(m_1) \Delta t(m_2)$ distances between high-dimensional features. Hence, in this paper we examine the computation of the distance between tracks $X(m_1)$ and $X(m_2)$ as the distance between their fixed-size representations. Yang et al. [5] proposed the 2-layer neural network with attention blocks to aggregate the CNN features of all frames. However, in our experiments the robustness of this approach was insufficient, hence, we use straight-forward aggregation (or pooling [5]) techniques:

1. The distance between tracks is defined as the distance between their medoids:

$$\rho(X(m_1), X(m_2)) = \rho(\mathbf{x}^*(m_1), \mathbf{x}^*(m_2)),$$

$$\mathbf{x}^*(m_i) = \operatorname{argmin}_{x(t), t \in [t_1(m_i), t_2(m_i)]} \sum_{t' = t_1(m_i)}^{t_2(m_i)} \rho(x(t), x(t')), i \in \{1, 2\}. \quad (2)$$

2. Average features of each track are matched:

$$\rho(\mathbf{X}(m_1), \mathbf{X}(m_2)) = \rho(\bar{x}(m_1), \bar{x}(m_2)), \bar{x}(m_i) = \frac{1}{\Delta t(m_i)} \sum_{t'=t_1(m_i)}^{t_2(m_i)} \mathbf{x}(t). \quad (3)$$

It is worth noting that in static image recognition tasks the CNN bottleneck feature vectors are typically divided into their L₂ norm [12]. Such normalization is known to make these features more robust to variations of observation conditions, e.g., camera resolution, illumination and occlusion. However, in our task the sequences of frames are matched, so it is possible to slightly defer the normalization. Thus, in this paper we consider either conventional approach with aggregation of the normalized features (hereinafter “L₂-norm -> Medoid” (2) and “L₂-norm -> AvePool” (3)), or its slightly modified version with normalization of aggregated vectors (2), (3) (hereinafter “Medoid -> L₂-norm” and “AvePool -> L₂-norm”, respectively).

We implemented the described approach in a special in MS Visual Studio 2015 project (github link will be provided after double-blind peer review) using C++ language and the OpenCV library, especially, its DNN and Tracking extra modules. The complete data flow in this system is presented in Fig. 1.

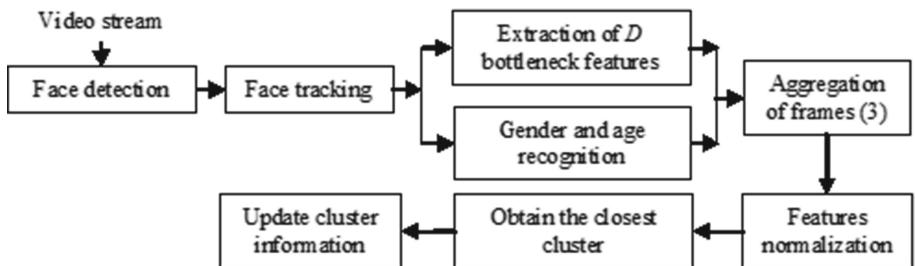


Fig. 1. The data flow in the organizing video data system.

Here we detect faces using the Viola-Jones cascades with the Haar features. The obtained facial regions are verified by additional eye detection [9] and tracked using the KCF algorithm [10]. Face detection is repeated periodically in order to: (1) verify the tracking results; (2) look for new faces, and (3) mark disappeared persons. In the latter case we extract D CNN bottleneck features for each frame of the track. These features are extended with the probabilities

at the output of the gender and age prediction CNNs [13]. In the data flow (Fig. 1) the simple online clustering of the normalized features is presented [14]: the feature vector of the last track is matched with the features of previously detected clusters. If the distance to the nearest cluster does not exceed a certain threshold, this track is added to the cluster, and the information about the latter is updated. Namely, we compute the aggregated features (2) or (3) of a whole cluster and re-estimate the gender and age of persons discovered in the input video in order to facilitate the navigation through the organized collection of tracks.

3 Experimental Results

In this section we provide experimental study of the key part of the proposed system (Fig. 1), namely, the matching of video tracks in unconstrained face verification task. In addition to described aggregation techniques (“ L_2 -norm -> Medoid” and “Medoid -> L_2 -norm” (2), “ L_2 -norm -> AvePool” (3)) we examined the pairwise comparison of all frames in both tracks (1). Moreover, we implemented the described techniques (1)–(3) unnormalized features. To extract features, we used the Caffe framework and two publicly available CNNs suitable for face recognition, namely, the VGGNet [6] and Lightened CNN (version C) [12]. The VGGNet extracts $D = 4096$ non-negative features in the output of “fc7” layer from 224×224 RGB images. $D = 256$ features (“eltwise_fc2” layer) are computed when 128×128 grayscale image of the facial region is fed into the Lightened CNN. All experiments were performed on the computer Lenovo ideapad 310, 64-bit operating system with NVIDIA GeForce 920MX.

Our first experiments were conducted on the YouTube Faces (YTF) database [15], which contains 3,425 videos of 1,595 different people. An average of 2.15 videos are available for each subject. The shortest track duration is 48 frames, the longest track contains 6,070 frames, and the average length of a video clip is 181.3. The estimates of AUC (Area under curve) and FRR (False Reject Rate) for fixed FAR (False Accept Rate) using the YTF face verification protocol are presented in Table 1 (in the format mean \pm standard deviation). Here we do not display a row for “ L_2 -norm -> Medoid”, because its results are identical to the “Medoid -> L_2 -norm” due to the independence of the computed medoid (2) on the order of normalization.

These results emphasize the need for proper normalization of feature vectors. The most efficient algorithm is to normalize the features of all frames and then find the average distance (1). The obtained state-of-the-art result is 0.982, the difference between it and 0.988 [5] is not statistically significant. However, the normalization of AvePool (3) features is characterized by practically the same quality, though it is much faster. The AUC for matching of medoids (2) is 10–12% less when compared to the AUC of the AvePool (3). It is worth noting that the latter method is also 1–2% more accurate than the conventional approach [4,5] of averaging the preliminarily normalized features.

In the next experiment we obtained a cluster threshold by fixing FAR = 1% and using training set, then applied the clustering of all tracks from the

Table 1. Results of video-based face verification, YTF dataset

	Lightened CNN			VGGNet		
	AUC (%)	ERR (%)	FRR@FAR = 1%	AUC (%)	ERR (%)	FRR@FAR = 1%
Distance (1)	90.7 ± 0.6	15.7 ± 0.2	77.0 ± 8.4	83.3 ± 0.8	24.0 ± 0.3	85.8 ± 9.0
L ₂ -norm -> Distance (1)	98.2 ± 0.4	6.0 ± 0.1	14.1 ± 3.6	97.9 ± 0.6	6.0 ± 0.1	23.2 ± 6.3
Medoid (2)	84.7 ± 0.7	25.0 ± 0.3	72.9 ± 7.8	80.8 ± 1.2	27.0 ± 0.4	83.9 ± 7.7
Medoid (2) -> L ₂ -norm	88.8 ± 0.6	19.0 ± 0.2	54.1 ± 5.9	85.2 ± 0.7	23.0 ± 0.2	69.9 ± 7.9
AvePool (3)	91.8 ± 1.4	13.0 ± 0.1	72.3 ± 11.5	87.4 ± 1.2	39.0 ± 0.3	81.2 ± 5.8
L ₂ -norm -> AvePool (3)	96.8 ± 0.5	12.0 ± 0.1	37.2 ± 7.6	96.3 ± 0.7	39.0 ± 0.3	76.9 ± 6.8
AvePool (3) -> L ₂ -norm	97.6 ± 0.5	7.5 ± 0.1	12.5 ± 3.1	97.7 ± 0.6	13.0 ± 0.1	25.3 ± 7.8

YTF. By using the Lightened CNN features, 1800 clusters were identified, 20 of them contain videos of different persons. The application of the VGGNet feature extraction increases the number of clusters to 2000 with 30 incorrect clusters.

The first experiment was repeated for rough grouping of tracks with the persons of approximately identical age and same gender using the probabilities at the outputs of the pre-trained CNNs [13]. Table 2 contains AUC achieved for matching of $D = 8$ posterior probabilities of age categories, $D = 2$ ((male/female) posterior probabilities) and the union of these two feature sets. We used L₁-norm to treat the features as posterior probabilities and compared them with either Euclidean (L₂) distance of the Kullback-Leibler (KL) divergence, which is assumed to be more suitable for comparison of discrete probability distributions. Here the prior feature normalization is not needed, as the outputs of the CNNs softmax layers are L₁ normed. These results are much worse when compared to facial features from the previous experiment. Nevertheless, the age and gender features can be potentially used to refine the results obtained by conventional face verification techniques (Table 1) AUC is 8-19% higher than the random guess.

Table 2. AUC (%) of video-based face verification, YTF dataset, age and gender features

	Distance	Age	Gender	Age and Gender
Distance (1)	L ₂	60.8 ± 1.3	65.8 ± 0.8	68.7 ± 0.8
	KL	61.6 ± 1.1	65.8 ± 1.0	65.8 ± 0.9
Medoid (2)	L ₂	58.4 ± 1.4	63.3 ± 1.0	64.9 ± 1.0
	KL	58.9 ± 1.4	63.4 ± 1.0	64.8 ± 0.9
AvePool (3) -> L ₂ -norm	L ₂	60.4 ± 1.3	65.7 ± 0.9	67.9 ± 0.9
	KL	63.2 ± 1.2	65.3 ± 0.9	68.8 ± 0.8

The last experiment was conducted on the IARPA Janus Benchmark A (IJB-A) (IJB-A) dataset [16] with 2043 videos of 500 identities. Table 3 contains the results of several best aggregation techniques in the face verification with bottleneck features extracted by VGGNet and Lightened CNN.

Table 3. Results of video-based face verification, IJB-A dataset

	Lightened CNN			VGGNet		
	AUC (%)	ERR (%)	FRR@FAR = 1%	AUC (%)	ERR (%)	FRR@FAR = 1%
L_2 -norm -> Distance (1)	87.9 \pm 0.5	20.5 \pm 0.9	67.9 \pm 3.3	97.5 \pm 0.4	8.0 \pm 0.4	30.3 \pm 4.6
Medoid (2) -> L_2 -norm	76.6 \pm 0.4	30.0 \pm 1.2	77.1 \pm 4.0	92.4 \pm 0.7	15.5 \pm 0.6	50.0 \pm 6.9
L_2 -norm -> AvePool (3)	79.6 \pm 0.7	27.8 \pm 0.8	67.5 \pm 4.4	96.1 \pm 0.4	13.6 \pm 0.8	40.0 \pm 4.4
AvePool (3) -> L_2 -norm	88.2 \pm 0.4	20.0 \pm 0.4	59.3 \pm 2.9	97.7 \pm 0.3	8.0 \pm 0.3	26.8 \pm 4.2

In contrast to the first experiment, here the VGGNet [6] is much more accurate than the Lightened CNN [12]. Our conclusions about relative efficiency of discussed aggregation techniques remain similar to the previous experiments. However, this dataset highlights the superiority of the normalized average features ($\text{AvePool} \rightarrow L_2$ -norm): it drastically improves AUC and FRR, when compared to traditional implementation of average pooling in aggregation of video features [3, 4].

4 Conclusion

In this paper we considered the automatic organizing the data in video surveillance systems (Fig. 1). We particularly focused on the ways to efficiently compute the dissimilarity of video tracks by using rather simple aggregation techniques. We experimentally supported the claim that the most accurate and computationally cheap technique involves the L_2 -normed average vector of unnormalized frame features. It was noticed that the sequence of this two operations is very important. In fact, much more widely used aggregation of normalized features [2] is usually less accurate (Table 3).

The main direction for further research is applying our approach in organizing data from real video surveillance systems. It is also important to examine more sophisticated distances between video tracks, e.g., metric learning [17] or statistical homogeneity testing [11]. If the number of observed persons is high, it is necessary to deal with insufficient performance of our simple online clustering by using, e.g., approximate nearest neighbor search [7, 8, 14, 18]. Moreover, we are planning to introduce the weighing for different features including age and gender probabilities to make our algorithm more accurate. In fact, the accuracy of the age and gender prediction CNNs [13] is rather low, hence, it is necessary to implement contemporary CNN architectures including Inception or ResNets.

Acknowledgements. The article was prepared within the framework of the Academic Fund Program at the National Research University Higher School of Economics (HSE) in 2017 (grant 17-05-0007) and by the Russian Academic Excellence Project “5–100”. Andrey V. Savchenko is partially supported by Russian Federation President grant no. MD-306.2017.9.

References

1. Manju, A., Valarmathie, P.: Organizing multimedia big data using semantic based video content extraction technique. In: IEEE International Conference on Soft-Computing and Networks Security (ICSNS), pp. 1–4 (2015)
2. Zhang, Y.J., Lu, H.B.: A hierarchical organization scheme for video data. *Pattern Recognit.* **35**(11), 2381–2387 (2002)
3. Chen, J.C., Ranjan, R., Kumar, A., Chen, C.H., Patel, V.M., Chellappa, R.: An end-to-end system for unconstrained face verification with deep convolutional neural networks. In: IEEE International Conference on Computer Vision Workshops, pp. 118–126 (2015)
4. Li, H., Hua, G., Shen, X., Lin, Z., Brandt, J.: Eigen-PEP for video face recognition. In: Cremers, D., Reid, I., Saito, H., Yang, M.-H. (eds.) ACCV 2014. LNCS, vol. 9005, pp. 17–33. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16811-1_2
5. Yang, J., Ren, P., Chen, D., Wen, F., Li, H., Hua, G.: Neural aggregation network for video face recognition (2016). [arXiv: 1603.05474](https://arxiv.org/abs/1603.05474)
6. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: Proceedings of the British Machine Vision, pp. 6–17 (2015)
7. Savchenko, A.V.: Deep convolutional neural networks and maximum-likelihood principle in approximate nearest neighbor search. In: Alexandre, L.A., Salvador Sánchez, J., Rodrigues, J.M.F. (eds.) IbPRIA 2017. LNCS, vol. 10255, pp. 42–49. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58838-4_5
8. Savchenko, A.V.: Deep neural networks and maximum likelihood search for approximate nearest neighbor in video-based image recognition. *Opt. Mem. Neural Netw. (Inf. Opt.)* **26**(2), 129–136 (2017)
9. Szeliski, R.: Computer Vision: Algorithms and Applications. Springer Science and Business Media, Berlin (2010)
10. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7575, pp. 702–715. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33765-9_50
11. Friedman, J., Hastie, T., Tibshirani, R.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics, New York (2009). <https://doi.org/10.1007/978-0-387-84858-7>
12. Wu, X., He, R., Sun, Z.: A lightened CNN for deep face representation (2015). [arXiv:1511.02683](https://arxiv.org/abs/1511.02683)
13. Levi, G., Hassner, T.: Age and gender classification using convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 34–42 (2015)
14. Savchenko, A.V.: Clustering and maximum likelihood search for efficient statistical classification with medium-sized databases. *Opt. Lett.* **11**(2), 329–341 (2017)
15. Wolf, L., Hassner, T., Maoz, I.: Face recognition in unconstrained videos with matched background similarity. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 529–534 (2011)
16. Klare, B.F., Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., Grother, P., Mah, A., Jain, A.K.: Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark, A. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1931–1939 (2015)

17. Kulis, B.: Metric learning: a survey. *Found. Trends Mach. Learn.* **5**(4), 287–364 (2013)
18. Babenko, A., Lempitsky, V.: The inverted multi-index. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3069–3076. IEEE (2012)

Satellite Image Forgery Detection Based on Buildings Shadows Analysis

Andrey Kuznetsov^{1,2}  and Vladislav Mysnikov^{1,2} 

¹ Samara National Research University, Samara, Russia

kuznetsov.a@ssau.ru, vmyas@geosamara.ru

² Image Processing Systems Institute, Branch of the Federal Scientific Research Centre Crystallography and Photonics, Russian Academy of Sciences, Samara, Russia

Abstract. Satellite images are to be effectively protected nowadays. There are a lot of ways of changing image content to hide important information: resampling, copy-move, object replacement and other attacks. When these changes are applied to satellite data inclination angles of shadows can be also changed. We propose a new method for satellite image forgery detection based on the analysis of high buildings shadows inclination angles on high resolution snapshots (0.5 m and less). In the proposed solution, the shadows are detected using Canny edge detector with further edge tracing. The comparison of both edge detection methods is presented in the experiments section. The next step is shadows inclination angles estimation using special model-oriented descriptors. The experiments show high accuracy of changed areas detection.

Keywords: Satellite image · Forgery · High buildings · High resolution Shadow detection · Inclination angle · Canny edge detector · Tracing

1 Introduction

Remote sensing data are widely used in modern world. They consist of two main components: a digital image and metadata, which describes the conditions of image creation process. During satellite data transmission from source to destination, these data can be distorted accidentally (due to errors) and artificially (by malicious users). When this happens, the satellite image itself or its metadata can be changed. The problem of forgery detection in digital images without additional information about the process of image creation is solved in [1–4].

Nowadays, the analysis of light parameters inconsistency for local parts of a single object in digital image is carried out. That solution was based on the usage of vector data of the snapshot territory to estimate the inclination angle of shadows of buildings. Moreover, satellite images metadata and vector maps of images territory allow to analyze the consistency of objects and their shadows. The analysis of papers showed that there is a little number of works aimed at the detection of inconsistency in shadows and objects in satellite images using a priori data or not.

In this paper, we propose a new solution for forgery detection in high resolution satellite images based on the analysis of high buildings shadows. The first stage of the

algorithm is edge detection with further edge tracing to detect the buildings and their corresponding shadows. During the second stage inclination angles of shadows are estimated and artificially changed areas are detected. Experimental results showed that the proposed solution has high accuracy and can be used as a forgery detection step in sophisticated software for remote sensing data analysis.

2 Proposed Solution

If we have no additional information about the territory of the snapshot we need to detect buildings and corresponding shadows using only image analysis methods. We will use high resolution images as the object of analysis. In this paper, we propose the algorithm that allows to identify the corresponding buildings corners and shadows of these corners using Canny detector [5]. This method provides precise edge detection results for noisy images and the edges are one pixel in width, which enables to trace them on the next step [6].

Let $f(m, n), m \in [0, M], n \in [0, N]$ be an analyzed satellite image (see Fig. 1), where M, N are image linear dimensions.



Fig. 1. Geoeye satellite image of Samara and its fragment

Firstly, we execute some preprocessing operations:

- convert the image to grayscale (if it is multichannel);
- filter noise to smooth the edges.

After that we need to detect edges to find the buildings and shadows corresponding to them. To solve this task, we apply Canny edge detection algorithm to the preprocessed image. The edge detection result is stored in the image $f'(m, n)$. We use two parameters of Canny edge detection method: the first one is used to select the most

significant boundaries (th_1), the second one is used to combine edge segments into contours (th_2). In our solution, we applied empirically selected parameters for edge detection $th_1 = 50, th_2 = 120$. These values lead to the best precision for edge detection. The result of Canny edge detector will be denoted as $c_f'(m, n)$.

The next step of the algorithm is line tracing of $c_f'(m, n)$ for image verification. It consists of two steps:

1. detection of corresponding angles of buildings and their shadows;
2. detection of shadows edges parts that are collinear with shadow inclination angle, calculated using the values of analyzed image metadata.

The first step of the proposed shadow detection algorithm is to detect angles between the edges close to 90° . The proximity measure of these values will be determined by a threshold parameter $\Delta_{rightAngle}$. Each building has a square corner of the roof, which corresponds to a square corner of its shadow. For each edge pixel (x_b, y_b) we execute eight-connected tracing procedure [6] in opposite directions and estimate the angle γ between these traced edge parts. If the following condition $\gamma \in [\frac{\pi}{2} - \Delta_{rightAngle}, \frac{\pi}{2} + \Delta_{rightAngle}]$ is satisfied, then (x_b, y_b) point is added to the list of points, which expect to be a building roof angle or a shadow angle. Then the points list is filtered to select the points $(x_1, y_1), (x_2, y_2)$ which satisfy the following conditions:

$$\left| arctg2\left(\frac{y_1 - y_2}{x_1 - x_2}\right) - \alpha_s \right| < \Delta_s, \quad (1)$$

$$height_{min} < \sqrt{(y_1 - y_2)^2 + (x_1 - x_2)^2} < height_{max}, \quad (2)$$

where α_s is the shadow angle, calculated from satellite image metadata, Δ_s is the threshold for angle deviation, $height_{min}$ and $height_{max}$ are the values of buildings minimum and maximum height.

The result of detection of buildings roofs and their shadows corresponding angles is presented in Fig. 2 (a part of Geoeye image).

The second step of the proposed algorithm is to identify edges of the shadows, which direction coincides with shadow inclination angle, calculated using the values of analyzed image metadata. In the basis of this operation also lies the tracing of $c_f'(m, n)$. Let K_s be a restriction on the maximum pixel length of the traced edge. When we determine a list of K_s points for a given point, we define a line using *Line2DFitting* function of *OpenCV*. We then obtain a point (x_{line}, y_{line}) belonging to this line and line direction vector (d_x, d_y) . The result of this operation is presented in Fig. 3.

In this case, the following problem arises in Fig. 3 (dark gray color indicates all the detected areas for which the approximated straight lines have a direction close to the metadata shadow angle). It can be seen from the figure that two curved segments are false detected as correct shadow borders (in Fig. 3 highlighted in red curved and linear segments are for which are detected as having inclination angle close to the true shadow angle). To solve the problem, we need some post processing procedure to filter false detected shadow borders. We propose the following solution. All the detected

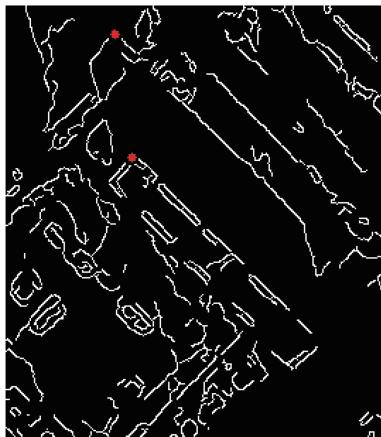


Fig. 2. Corresponding angles detection for a building

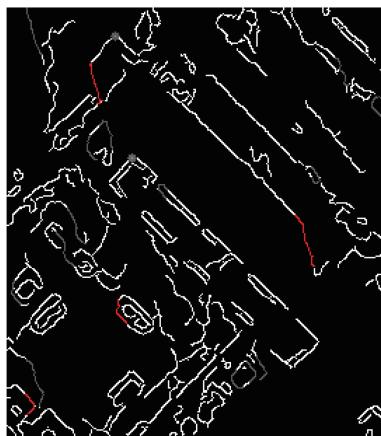


Fig. 3. Shadow borders detection result (Color figure online)

segments are divided into k sequential parts and each of them is approximated as a line. If for k shadow border parts there are $\frac{k}{2}$ pieces, which have inclination angle more than Δ_s , that such a segment is removed from the list of potential shadow borders of buildings. After this post processing procedure, the result of shadow borders detection will have less false detected segments (see Fig. 4).

Having these shadow segments detected we can estimate the shadow inclination angle value and compare it with the valid value from metadata. One approach is to use model-oriented descriptor (MOD) to estimate the angle value [7]. It is calculated for the neighborhood of the shadow border. If the building has close inclination angle to the valid value, the descriptor value is close to 1. If the angle differs from the valid for at least 5° , the descriptor value will be close to 0.

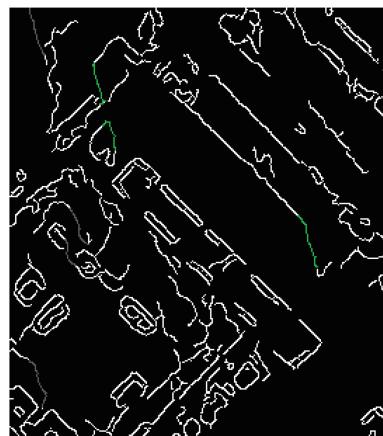


Fig. 4. Shadow borders detection result after filtering

3 Experiments

To conduct an experiment, we take a Geoeye-1 satellite image (0.5 m resolution) and insert in it rotated buildings (20 objects) copied from the same image. Rotation leads to appearance of buildings with wrong shadows directions. Considering the randomness of objects location, some shadow borders may intersect with shadows of other objects.

The valid angle for the test satellite image is 76° . Then we copy and paste 20 objects with changed inclination angle from 0 to 360° . The result of validation procedure based on shadow borders detection and estimation of the inclination angle is presented in Fig. 5. For shadow inclination angle from 0 to 65 and from 85 to 360° most of inserted objects are detected.

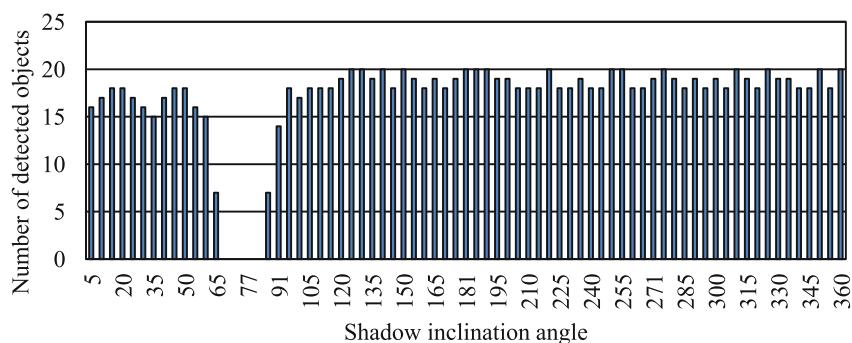


Fig. 5. Dependency of test sample objects number that failed validation test from shadow inclination angle.

4 Results and Conclusion

We proposed a new algorithm for satellite image forgery detection, when some malicious user adds some image fragment with incorrect shadow inclination angle to hide some important information. The proposed solution provides detection of image fragments with incorrect shadow inclination angles analyzing the buildings shadows with Canny edge detector and further edge tracing. Experimental results showed high accuracy of the proposed method for shadow inclination angle deviation more than 10°. Further we will carry out research in decreasing this deviation.

References

1. Christlein, V., Riess, C., Jordan, J., Riess, C., Angelopoulou, E.: An evaluation of popular copy-move forgery detection approaches. *IEEE Trans. Inf. Forensics Secur.* **7**(6), 1841–1854 (2012)
2. Farid, H.: Exposing digital forgeries from JPEG ghosts. *IEEE Trans. Inf. Forensics Secur.* **1**(4), 154–160 (2009)
3. Farid, H.: Image forgery detection. *IEEE Signal Process. Mag.* 16–25 (2009)
4. Vladimirovich, K.A., Valerievich, M.V.: A fast plain copy-move detection algorithm based on structural pattern and 2D Rabin-karp rolling hash. In: Campilho, A., Kamel, M. (eds.) *ICIAR 2014. LNCS*, vol. 8814, pp. 461–468. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11758-4_50
5. Canny, J.: A computational approach to edge detection. *Pattern Anal. Mach. Intell. IEEE Trans. PAMI* **8**(6), 679–698 (1986)
6. Ren, M., Yang, J., Sun, H.: Tracing boundary contours in a binary image. *Image Vision Comput.* **20**(2), 125–131 (2002)
7. Myasnikov, V.: Model-based gradient field descriptor as a convenient tool for image recognition and analysis. *Comput. Opt.* **36**(4), 596–604 (2012)

Nonlinear Dimensionality Reduction of Hyperspectral Data Using Spectral Correlation as a Similarity Measure

Evgeny Myasnikov^(✉)

Samara University, 34, Moskovskoye shosse, Samara 443086, Russia
mevg@geosamara.ru
<http://www.ssau.ru>

Abstract. In this paper, we propose a novel dimensionality reduction method, which is based on the principle of preserving the pairwise spectral correlation measures. For the proposed method, we introduce the corresponding quality measure, and derive the numerical optimization algorithm based on a stochastic gradient descent technique. We provide the results of the experimental study that compares the method to the principal component analysis method using well-known hyperspectral scenes. The results of the study show that the proposed method can be successfully applied to process hyperspectral images.

Keywords: Hyperspectral image · Spectral correlation
Nonlinear dimensionality reduction · Nonlinear mapping
Principal component analysis

1 Introduction

Despite a huge number of works devoted to the processing and analysis of hyperspectral data, a relatively small number of works are devoted to a nonlinear dimensionality reduction of hyperspectral data. Among such works it is possible to indicate the works, which use locally linear embedding (LLE) [1], laplacian eigenmaps (LE) [2], isometric embedding (ISOMAP) [3], nonlinear mapping (NLM) [4] and curvilinear component analysis (CCA) [5] as particular nonlinear dimensionality reduction methods. All these methods require to specify the measure of dissimilarity between pixels in a hyperspectral image. The most frequently used measure for this purpose is Euclidean distance. Although we can indicate a few studies [3, 7, 8] in which spectral angle mapper (SAM) measure was used with nonlinear dimensionality reduction methods, other dissimilarity measures remain poorly studied.

In this context, the spectral correlation measure [9] is of particular interest, as it proved to have advantages [10] over the Euclidean distance and spectral angle mapper measure in the field of hyperspectral image analysis.

In this paper, we propose a novel nonlinear dimensionality reduction method based on the principle of preserving the pairwise spectral correlation measures between pixels of a hyperspectral image.

The paper is organized as follows. In the next section we introduce the objective function, and derive a nonlinear dimensionality reduction method based on the principle of preserving the pairwise spectral correlation measures. In Sect. 3, we describe the numerical optimization algorithm that can be applied to hyperspectral images. Section 4 describes the experimental studies. The paper ends with conclusions.

2 Nonlinear Dimensionality Reduction Based on Spectral Correlation Measure

Let us consider the hyperspectral image X of width W and height H . This image contains $N = W * H$ pixels $x_i, i = 1..N$. Each pixel of the image X can be considered as a vector in the M -dimensional hyperspectral space R^M . In this paper, we study the spectral correlation measure [9]. This measure is expressed as the following:

$$r(x_i, x_j) = \frac{K \sum_{k=1}^K x_{ik} x_{jk} - \sum_{k=1}^K x_{ik} \sum_{k=1}^K x_{jk}}{\sqrt{K \sum_{k=1}^K x_{ik}^2 - (\sum_{k=1}^K x_{ik})^2} \sqrt{K \sum_{k=1}^K x_{jk}^2 - (\sum_{k=1}^K x_{jk})^2}}. \quad (1)$$

Here x_i and x_j are hyperspectral image pixels, K is the number of overlapping spectral bands.

This measure is often written in another form, which is known as a Pearson correlation coefficient:

$$r(x_i, x_j) = \frac{\sum_{k=1}^K (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{\sqrt{\sum_{k=1}^K (x_{ik} - \bar{x}_i)^2} \sqrt{\sum_{k=1}^K (x_{jk} - \bar{x}_j)^2}}, \quad (2)$$

where \bar{x} is the mean of x .

In this paper, we derive a dimensionality reduction method based on the principle of preserving the pairwise spectral correlation measures between pixels of a hyperspectral image. To do this let us introduce the following quality measure of spectral correlation preservation:

$$\varepsilon_r = \mu \sum_{i,j=1(i < j)}^N (r(x_i, x_j) - r(y_i, y_j))^2. \quad (3)$$

Here x_i are pixels of the source hyperspectral image, and y_i are corresponding vectors in a reduced low-dimensional space R^m . The normalizing factor

$$\mu = \frac{1}{\sum_{i,j=1(i < j)}^N r^2(x_i, x_j)} \quad (4)$$

is a constant for a given image.

This measure (3) shows how well pairwise spectral correlation measures (1) are preserved by the dimensionality reduction procedure.

As we would like to minimize the quality measure (3), we solve the following optimization problem:

$$\varepsilon_r \rightarrow_Y \min. \quad (5)$$

Here Y is the parameter set, which consists of all the vectors in the reduced low-dimensional space R^m : $Y = (y_1, \dots, y_N)$.

To find optimum values for the optimized parameters Y , we use a gradient based algorithm as the most often used technique in similar cases.

To solve the problem (5), this algorithm starts with some initial configuration $Y(0)$, and then sequentially narrows the optimized parameters using the following equation:

$$Y(t+1) = Y(t) - \alpha \nabla \varepsilon_r. \quad (6)$$

Here t is a number of an iteration, $\nabla \varepsilon_r$ is a gradient of the objective function (3), α is a coefficient of the gradient descent.

Now let us calculate the partial derivatives of the quality measure (3):

$$\frac{\partial \varepsilon_r}{\partial y_{ik}} = \mu \sum_{j=1(j \neq i)}^N \left(-2(r(x_i, x_j) - r(y_i, y_j)) \frac{\partial r(y_i, y_j)}{\partial y_{ik}} \right) \quad (7)$$

Here the partial derivatives of the spectral correlation measure (1) can be written as:

$$\frac{\partial r(y_i, y_j)}{\partial y_{ik}} = \frac{1}{g_{ij}} \left(my_{jk} - \sum_{l=1}^m y_{jl} - \left(my_{ik} - \sum_{l=1}^m y_{il} \right) f_{ij} \right) \quad (8)$$

where

$$f_{ij} = \frac{m \sum_{k=1}^m y_{ik} y_{jk} - \sum_{k=1}^m y_{ik} \sum_{k=1}^m y_{jk}}{m \sum_{k=1}^m y_{ik}^2 - (\sum_{k=1}^m y_{ik})^2}, \quad (9)$$

$$g_{ij} = \sqrt{m \sum_{k=1}^m y_{ik}^2 - \left(\sum_{k=1}^m y_{ik} \right)^2} \sqrt{m \sum_{k=1}^m y_{jk}^2 - \left(\sum_{k=1}^m y_{jk} \right)^2}. \quad (10)$$

Finally, the recurrent equation for the coordinates of the output vectors in the reduced space takes the following form:

$$y_{ik}(t+1) = y_{ik}(t) + 2\alpha\mu \sum_{j=1}^N \frac{r(x_i, x_j) - r(y_i, y_j)}{g_{ij}} \left(my_{jk} - \sum_{l=1}^m y_{jl} - \left(my_{ik} - \sum_{l=1}^m y_{il} \right) f_{ij} \right). \quad (11)$$

Thus, the proposed dimensionality reduction procedure reduce a dimensionality based on the spectral correlation preserving principle. The proposed method can be considered as an analogue of the approaches [11, 12] for the spectral correlation measure. A similar technique for the spectral angle measure can be found in [8].

3 Numerical Optimization Algorithm

The described above dimensionality reduction method cannot be applied to hyperspectral remote sensing images directly due to the high computational complexity and memory limitations of this method. It is required to execute $O(MmN^2)$ operations per one iteration of the optimization process. Thus, the computational complexity depends on the dimensionality of input (M) and output (m) spaces, and the number of pixels (N). Optionally, it is possible to calculate and store the matrix of input spectral correlation measures $\{r(x_i, x_j)\}$, but it takes $N(N - 1)/2$ floating point values. To overcome this problem, we adopted stochastic gradient descent algorithm. For this algorithm, the value of the gradient $\nabla \varepsilon_r$ in the Eq. (6) is estimated using a random subsample [13]. In this case, the Eq. (11) takes the following form:

$$y_{ik}(t+1) = y_{ik}(t) + 2\alpha\mu \sum_{j=1}^S \frac{r(x_i, x_{s_j}) - r(y_i, y_{s_j})}{g_{is_j}} \left(my_{s_j k} - \sum_{l=1}^m y_{s_j l} - \left(my_{ik} - \sum_{l=1}^m y_{il} \right) f_{is_j} \right). \quad (12)$$

where s is a random subsample (mini-batch) used to approximate the gradient at the iteration t of the optimization process, s_j is the j -th element of this subsample, S is the cardinality of the subset s . Using this approach, the cardinality of mini-batch determines the computational complexity of the algorithm per one iteration. Thus, for subsets of size S , the computational complexity is reduced to $O(MmSN)$.

Finally, the overall numerical optimization algorithm can be expressed as the following:

1. Initialize $y_i(0), i = 1..N$ using the principal component analysis (PCA) technique;
2. While the stop criterion is not met:
 - (a) Generate the random subsample s used to approximate the gradient,
 - (b) Update each vector $y_i(t), i = 1..N$ using (12),
 - (c) (Optionally) Estimate the intermediate value of the objective function (3) using a random subsample;
3. Estimate the final value of the objective function (3).

4 Experimental Results

In this section, we present the results of the experimental study of the proposed nonlinear dimensionality reduction method. The experiments were conducted on the well-known Indian Pines Test Site 3 hyperspectral image [14] (see Fig. 1a).

This image was acquired by the AVIRIS sensor in North-western Indiana. It contains 145×145 pixels containing 224 spectral reflectance bands.

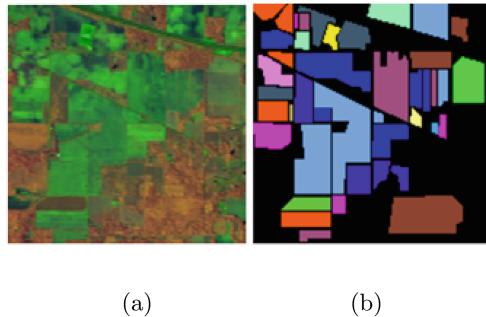


Fig. 1. Indian Pines Test Site 3 hyperspectral image: false color representation of the image produced using the nonlinear mapping technique (a), ground truth image (b), classified pixels are shown with colors (Color figure online).

At first, we estimated the quality indicator (3) and the average time per one iteration of the optimization algorithm. Some results of the study are presented in the Figs. 2a and 3. As it can be seen from the Fig. 2a the proposed method allows us to significantly reduce the error (3) compared to the initial value, which was obtained using the PCA technique. This indicates that the proposed technique operates properly, so that the pairwise spectral correlation measures (1) between vectors in output lower-dimensional space approximate corresponding spectral correlation measures in the source hyperspectral space.

As it can be seen from the Fig. 3a the final values of the error (3) decrease with the growth of the dimensionality of the output space. This observation is logical as it is easier to preserve similarity measures in higher-dimensional spaces. At the same time, the error values for dimensionality of the output space $m < 10$ are considerably higher than for $m \geq 10$.

The average time per one iteration in the Fig. 3b grows as a linear function of the output dimensionality. This is consistent with theoretical estimation of the computational complexity.

To compare the proposed method to similar alternative approaches we implemented the classical nonlinear dimensionality reduction algorithm [11, 12] using the stochastic gradient descent approach. In our implementation, we used $\delta(x_i, x_j) = 1 - r(x_i, x_j)$ as the dissimilarity measure between input vectors in the hyperspectral space. According to the base approach, we used the Euclidean

distance $d(x_i, x_j)$ as the dissimilarity measure in the output lower-dimensional space. Thus, this alternative approach minimizes the following error:

$$\varepsilon_{r \rightarrow d} = \frac{1}{\sum_{i,j=1(i < j)}^N \delta^2(x_i, x_j)} \sum_{i,j=1(i < j)}^N (\delta(x_i, x_j) - d(y_i, y_j))^2. \quad (13)$$

This approach provided considerably higher (an order of magnitude and greater) values of the error (13) compared to the corresponding values of (3) provided by the proposed method (compare Fig. 2(a), (b)).

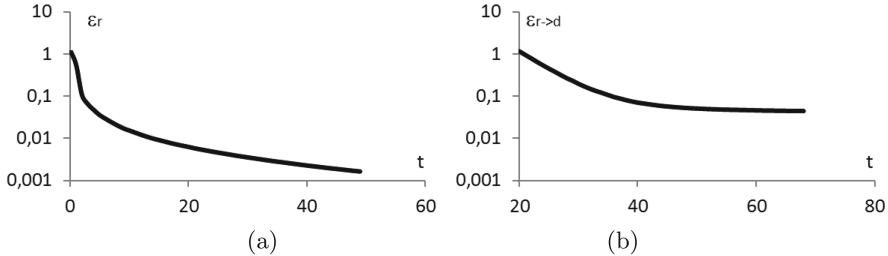


Fig. 2. The dependency of the quality measure (3) ε_r for the proposed method (a), and the quality measure (13) $\varepsilon_{r \rightarrow d}$ for the alternative approach (b) estimated using a random sample on the number of an iteration t (output dimensionality $m = 5$).

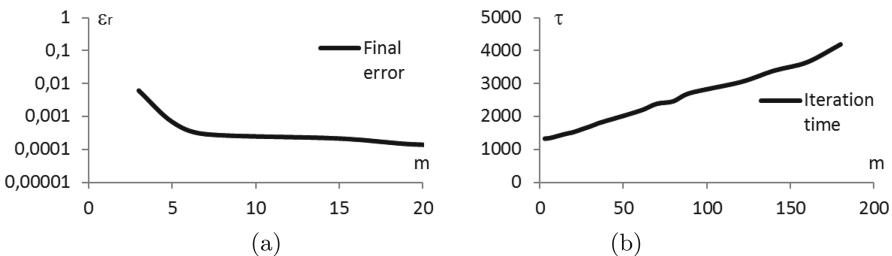


Fig. 3. The dependency of the quality measure (3) ε_r (a) and the average time τ per one iteration, in milliseconds (b), on the dimensionality (m) of the reduced space for the proposed method.

To indicate possible applications of the proposed method, we compared the classification quality of the nearest neighbor classifier using the features obtained by the PCA technique and by the proposed method. To perform the experiment, we used the ground-truth classification of the test image shown in the Fig. 1b to

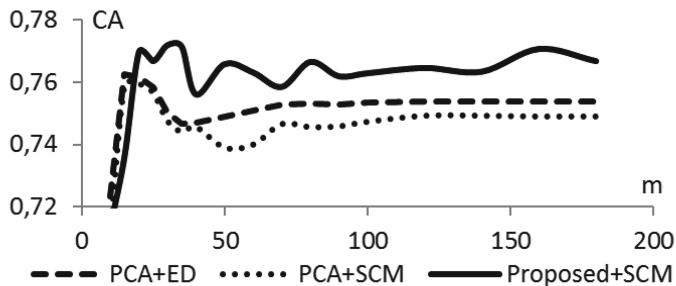


Fig. 4. The dependency of the classification accuracy (CA) on the dimensionality of the reduced space (m) for the proposed nonlinear dimensionality reduction method and the PCA technique with nearest neighbor classifier based on the Euclidean distance (ED) and spectral correlation measure (SCM).

compare the quality of the classification. We divided the whole set of 10249 classified ground-truth samples into a training subset and a test subset, consisting of 60 and 40% of samples respectively. The dimensionality of the reduced space ranged from 2 to 180. To assess the classification quality, we used the classification accuracy, which is defined as a proportion of correctly classified pixels of a test subset. The results of the experiment is shown in the Fig. 4.

As it can be seen from the results, the nearest neighbor classifier performed better for the features obtained using the proposed method, if the dimensionality of the output space was greater then 20. For lower dimensionality, the nearest neighbor classifier with PCA-based features provided better results.

5 Conclusion

In this paper, we proposed a novel dimensionality reduction method, which is based on the principle of preserving the pairwise spectral correlation measures. For the proposed method, we introduced the corresponding quality measure, and derived the numerical optimization algorithm based on the stochastic gradient descent technique. We conducted the experimental study, which showed that the proposed method significantly decrease the corresponding objective function, and can be successfully applied in hyperspectral data analysis.

In the near future, we plan to apply the similar approach to study other dissimilarity measures and consider exploiting spatial context in nonlinear mapping of hyperspectral image data.

Acknowledgments. This work is supported by Russian Foundation for Basic Research, project no. 16 – 37 – 00202 mol_a.

References

1. Kim, D.H., Finkel, L.H.: Hyperspectral image processing using locally linear embedding. In: First International IEEE EMBS Conference on Neural Engineering, pp. 316–319 (2003)
2. Shen-En, Q., Guangyi, C.: A new nonlinear dimensionality reduction method with application to hyperspectral image analysis. In: IEEE International Geoscience and Remote Sensing Symposium, pp. 270–273 (2007)
3. Bachmann, C.M., Ainsworth, T.L., Fusina, R.A.: Exploiting manifold geometry in hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **43**(3), 441–454 (2005)
4. Myasnikov, E.: Nonlinear mapping methods with adjustable computational complexity for hyperspectral image analysis. In: Proceedings SPIE 9875, p. 987508 (2015)
5. Lennon, M., Mercier, G., Mouchot, M., Hubert-Moy, L.: Curvilinear component analysis for nonlinear dimensionality reduction of hyperspectral images. In: Proceedings SPIE 4541, pp. 157–168 (2002)
6. Kruse, F.A., Boardman, J.W., Lefkoff, A.B., Heidebrecht, K.B., Shapiro, A.T., Barloon, P.J., Goetz, A.F.H.: The spectral image processing system (SIPS) - interactive visualization and analysis of imaging spectrometer data. *Remote Sens. Environ.* **44**, 145–163 (1993)
7. Yan, L., Niu, X.: Spectral-angle-based laplacian eigenmaps for nonlinear dimensionality reduction of hyperspectral imagery. *Photogram. Eng. Remote Sens.* **80**(9), 849–861 (2014)
8. Myasnikov, E.: Nonlinear mapping based on spectral angle preserving principle for hyperspectral image analysis. In: Felsberg, M., Heyden, A., Krüger, N. (eds.) CAIP 2017. LNCS, vol. 10425, pp. 416–427. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64698-5_35
9. Blumenson, L.E., Bakker, W.: CCSM: cross correlogram spectral matching. *Int. J. Remote Sens.* **18**(5), 1197–1201 (1997)
10. Kong, X., Shu, N., Huang, W., Fu, J.: The research on effectiveness of spectral similarity measures for hyperspectral image. In: 3rd International IEEE Congress on Image and Signal Processing (CISP2010), pp. 2269–2273 (2010)
11. Kruskal, J.B.: Multidimensional scaling by optimizing goodness of fit to a non-metric hypothesis. *Psychometrika* **29**, 1–27 (1964)
12. Sammon, J.W.: A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.* **18**(5), 401–409 (1969)
13. Myasnikov, E.: Evaluation of stochastic gradient descent methods for nonlinear mapping of hyperspectral data. In: Campilho, A., Karray, F. (eds.) ICIAR 2016. LNCS, vol. 9730, pp. 276–283. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41501-7_31
14. Hyperspectral Remote Sensing Scenes. http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes

Large-Scale Shape Retrieval with Sparse 3D Convolutional Neural Networks

Alexandr Notchenko^{1,2(✉)}, Yermek Kapushev^{1,2}, and Evgeny Burnaev¹

¹ Skolkovo Institute of Science and Technology, Skolkovo Innovation Centre,
Building 3, Moscow 143026, Russia

{alexandr.notchenko,yermek.kapushev}@skolkovotech.ru,
e.burnaev@skoltech.ru

² Institute for Information Transmission Problems RAS,
Bolshoy Karetny per. 19, build.1, Moscow 127051, Russia

Abstract. In this paper we present results of performance evaluation of S3DCNN — a Sparse 3D Convolutional Neural Network — on a large-scale 3D Shape benchmark ModelNet40, and measure how it is impacted by voxel resolution of input shape. We demonstrate comparable classification and retrieval performance to state-of-the-art models, but with much less computational costs in training and inference phases. We also notice that benefits of higher input resolution can be limited by an ability of a neural network to generalize high level features.

Keywords: Deep Learning · Sparse 3D Convolutional Neural Network
Voxel resolution

1 Introduction

For computer vision systems a precise and robust operations in real environments is only possible by harnessing information from 3D data. To achieve this we need to overcome some challenges of this kind of problems.

Data, received from such devices as 2.5D scanners, is often given in the form of noisy meshes or point clouds, which is not the best fit for new kinds of models such as Convolutional Neural Networks (CNN's) [14].

In the current state-of-the-art systems Convolutional Neural Networks are widely used, their effectiveness at processing 2D images is also suggestive of their efficacy to process 3D objects if presented in the form of several rendered views of the object. For example on one ModelNet40 [22] benchmark three recent papers, based on this idea, showed incremental improvements in recognition performance [8, 10, 18]. However, it can be argued that high performance is predicated by the usage of CNNs pre-trained on ImageNet [6].

Voxel representation of 3D shapes (i.e. a shape is represented as a three-dimensional grid, where occupied cells are binary values) are compatible with ConvNets input layers but create a number of difficulties. Adding a third spatial dimension in the input grid correspondingly increases computational costs.

Number of cells scales as a power of three w.r.t. the resolution of the voxel grid. Low resolution grids make it difficult to differentiate between similar shapes, and lose some of the fine details available in 2D renderings of equivalent resolution.

Some 3D Dense Convolutional Networks have been evaluated on the ModelNet40 benchmark [3, 16, 17, 22], but they still not perform as well as their multi-rendering 2D counterparts.

At the same time using Modified Spatially Sparse Neural Networks algorithms [7] to process data we are able to have reasonable training and inference time even with input resolution up to 100^3 voxels.

In this work, we present Sparse 3D Deep Convolutional Neural Networks and explore their ability to perform large-scale shape retrieval on the popular benchmark ModelNet40 [22] depending on an input resolution and a network architecture.

Sparse 3D CNNs are able to generate relevant features for retrieval analogously to 2D extractors. To have a system that uses many 2D rendered projections for inference is computationally very costly, especially for the task of Large Scale 3D Shape Retrieval. In this paper we present some preliminary results of our attempt to find out if the resolution of an object Voxelization impacts on descriptive feature extraction as measured by the retrieval performance on a sufficiently big dataset. Also we demonstrate ability of Sparse 3D CNNs to perform metric learning in the triplet loss setup. Lastly we train our model to perform classification on the ModelNet40 benchmark.

In Sect. 2 we formulate the problem in more detail and discuss latest relevant methods. In Sect. 3 we describe our approach to neural networks that helps us to solve the problem posed in Sect. 2. In Sect. 4 we document conditions of computational experiments we performed. In Sect. 5 we discuss results and make conclusions about our approach to the problem.

2 3D Large-Scale Retrieval

2.1 Large-Scale 3D Shape Datasets

As can be seen the great improvements in recent years for the problem of 2D large-scale image recognition, are not just the result of wide-spread adoption of Deep Learning techniques, but also it is due to the availability of large datasets that capture sufficient variety of features at different scales to be representative of some domain. However, only recently in the 3D recognition and retrieval such datasets started being published.

The recent competition ModelNet evaluated several models utilizing Neural Networks for 3D retrieval. ModelNet40 is a subset of this dataset, and it is going to be our main benchmark for the retrieval task. The approach for creating descriptors from multiple projections of a 3D shape with a transfer learning from ImageNet showed the best performance [18]. No full 3D algorithms that process voxels directly have been described up to now.

2.2 Shape Descriptors

To make inferences about 3D objects for purposes of computer vision or computer graphics, researchers developed a big amount of shape descriptors [4, 11–13].

Shape descriptors usually fit into two categories: one where shape descriptors are computed using 3D representations of objects, e.g. voxel discretizations, meshes, point clouds, or implicit surfaces, and the second one that describes a shape of a 3D object by a collection of 2D projections, often from multiple viewpoints.

Before large-scale 3D shape datasets such as ModelNet [22] and 3dShapeNet model which learns shape descriptors from voxel representation of a mesh object through 3D convolutional nets, 3D shape descriptors were mostly special functions capturing specific geometric properties of the shape surface or volume, for example: spherical functions computed on volumetric grids [11], generalization of SIFT and SURF feature descriptors for voxel grids [12], or for non-rigid bodies and deformable shapes heat kernel signatures on meshes [4, 13]. Developing classifiers and other supervised machine learning algorithms on top of such 3D shape descriptors poses a number of challenges. The success of CNNs image descriptors allows us to hope that descriptors based on 3D convolutional nets can be also beneficial compared to classic descriptors.

2.3 Triplet Learning

Recent work in [9] shows that learning representations with triplets of examples gives much better results than learning with pairs using the same network. Inspired by this, we focus on learning feature descriptors based on triplets of patches.

Learning with triplets involves training from samples of the form (a, p, n) , where

- a is an anchor object,
- p denotes a positive object, which is a sample we want to be closer to a and usually being a different sample of the same class as p , and
- n is a negative sample belonging to a different class than a and p .

Optimizing parameters of the network brings a and p close in the feature space, and pushes apart a and n .

Finally, let us introduce this triplet loss, also known as the ranking loss. It was first proposed for learning embedding using CNNs in [21] and can be defined as follows:

- Let us define $\delta_+ = \text{cosine}(f(a), f(p))$ and $\delta_- = \text{cosine}(f(a), f(n))$, i.e. this is a cosine distance between some feature representations $f(\cdot)$ for different objects,
- Then for a particular triplet we calculate the triplet loss using the formula

$$\lambda(\delta_+, \delta_-) = \max(0, \mu + \delta_+ - \delta_-),$$

where μ is a margin parameter. The correct order should be $\delta_- > \delta_+ + \mu$,

- If order of objects, provided by their corresponding descriptors are incorrect w.r.t. the triplet loss, then the network adjusts its weights through back-propagation signal to reduce the error.

3 Sparse Neural Networks

Using sparsity to make a neural network computations more efficient is pioneered by Benjamin Graham [7], who developed a low-level C++/CUDA library SparseConvNet¹ that implements strided convolutions and max-pooling operations on a D -dimensional sparse tensors using GPU. Due to this inherited sparsity we are able to process data in reasonable training and inference time even with input resolution up to hundreds of voxels. More precisely an information about voxels in a given layer is not stored in a 3-dimensional array, but in a sparse vector with active cells as elements.

Transformation of data between layers (e.g. convolutions, pooling, nonlinear activation functions), are performed on those sparse vectors. Data in areas with inactive voxels, which are most of them, does not depend on a voxel relative position, therefore it can be replaced by vectors of a smaller size without explicit spatial dimensions.

It's well known that, operating with a sparse data structures is more efficient than working with dense data. Another useful property is that we need to store much less data for each object. We have computed sparsity for all classes of ModelNet40 train dataset at voxel resolution equal to 40, and it's only 5.5%.

Paper [22] describes using 3D convolutions for their deep model. Voxel labeled as active when it's intersects with a mesh object, and inactive otherwise. This binary representation of 3D shape given as input to a 3D CNN, which has a structure similar to a 2D one. The main problem of this approach is ineffectiveness with which data is represented and processed. Mentioned model uses 30^3 cells, which is approximately the number of pixels in 2D applications of CNN. If we take into account linear dimensions it's obviously not a lot, as can be seen from Fig. 2. That resolution was primarily chosen because of computational resource limitation. Besides that, — convolution is very computationally expensive operation, complexity of which rises very fast with input scale. Computational complexity of 3D convolution for image with dimensions of $N \times M \times K$ with filters sizes of $n \times m \times k$ is equal to $\mathcal{O}(NMKnmk)$. If we use Fast Fourier Transform (FFT), complexity can be reduced to $\mathcal{O}((N+n)(M+m)(K+k) \log((N+n)(M+m)(K+k)))$ in exchange for more memory cost [15]. But even in that case, complexity of convolutions makes it impossible to work with objects in big voxel resolutions.

3.1 PySparseConvNet

The SparseConvNet Library is written in C++ programming language, and utilizes a lot of CUDA capabilities for speed and efficiency. But it is very limited when it comes to

¹ <https://github.com/btgraham/SparseConvNet>.

- extending functionality — class structure and CUDA kernels are very complex, and require re-compilation on every modification,
- changing loss functions — the only learning configuration was SoftMax with log-likelihood loss function,
- fine grained access to layer activations — there was no way to extract activations and therefore features from hidden layers,
- interactivity for models exploration — every experiment had to be a compiled binary with no way to perform operations step by step, to explore properties of models.

Because of all these problems we developed PySparseConvNet². On implementation level it's a python compiled module that can be used by Python interpreter, and harness all of it's powerful features. Most of modern Deep Learning tools, such as [1, 19, 20], use Python as a way to perform interactive computing.

Interface of PySparseConvNet is much simpler, and consist's of 4 classes:

- **SparseNetwork** — Network object class, it has all the methods to change it's structure, manipulate weights and activations,
- **SparseDataset** — Container class for sparse samples and their labels,
- **SparseBatch** — Gives access to data in dataset when processing separate mini-batches,

Table 1. S3DCNN network architecture.

Layer #	Layer type	Size	Stride	Channels	Spatial size	Sparsity (%) ^a
0	Data input	-	-	1	126	0.18
1	Sparse convolution	2	1	8	125	-
2	Leaky ReLU ($\alpha = 0.33$)	-	-	32	125	0.35
3	Sparse MaxPool	3	2	32	62	0.69
4	Sparse convolution	2	1	256	61	-
5	Leaky ReLU ($\alpha = 0.33$)	-	-	64	61	1.07
6	Sparse MaxPool	3	2	64	30	1.93
7	Sparse convolution	2	1	512	29	-
8	Leaky ReLU ($\alpha = 0.33$)	-	-	96	29	3.26
9	Sparse MaxPool	3	2	96	14	7.32
10	Sparse convolution	2	1	768	13	-
11	Leaky ReLU ($\alpha = 0.33$)	-	-	128	13	15.14
12	Sparse MaxPool	3	2	128	6	46.30
13	Sparse convolution	2	1	1024	5	-
14	Leaky ReLU ($\alpha = 0.33$)	-	-	160	5	97.54
15	Sparse MaxPool	3	2	160	2	100.00
16	Sparse convolution	2	1	1280	1	-
17	Leaky ReLU ($\alpha = 0.33$)	-	-	192	1	100.00

^a Last column “sparsity” is computed for render size = 40 and averaged for all samples

² <https://github.com/gangiman/PySparseConvNet>.

- **Off3DPicture** — Wrapper class for 3D models in OFF (Object File Format), used to voxelize samples to be processed by SparseNetwork.

4 Experiments

4.1 ModelNet40 Dataset

In our experiments we used well known data set of 3D objects ModelNet40. It is a subset of 40 classes of larger data set called ModelNet [22] that contains different 3D CAD models in OFF format.

The total size of ModelNet40 data set 12311. The data set is split into training and test subsets, their sizes are 9843 and 2468 correspondingly. The data set is not balanced. Number of samples per class vary: from 64 to 889, see Fig. 1.

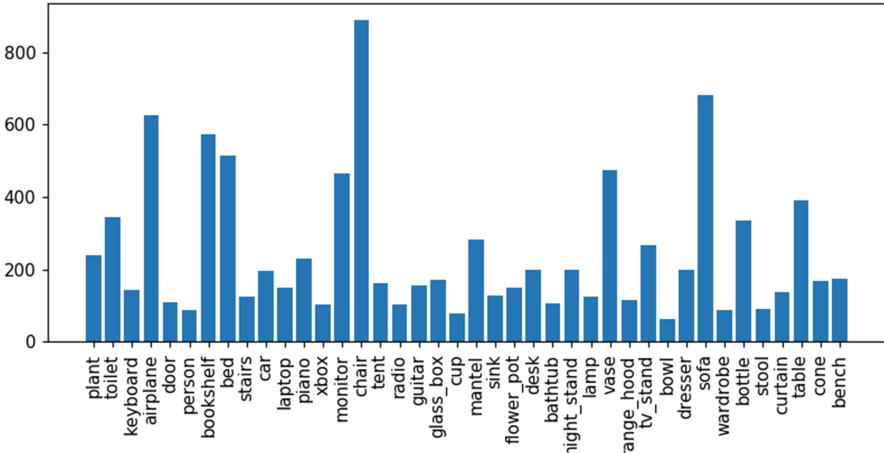


Fig. 1. ModelNet40 data set: distribution of samples per class.

4.2 Implementation Details

To demonstrate the impact that the triplet based training has on the performance of CNN descriptors we use a deep network architecture shown in a Table 1. This network was implemented in PySparseConvNet, which is our modification of the SparseConvNet library [7]. Besides new loss functions PySparseConvNet can be accessed from Python for a more interactive usage.

When forming a triplet for training we choose uniformly randomly a positive pair of objects from one class and select a negative sample uniformly randomly from one of other classes.

For the optimization we use the SGD [2], and the training is done

- in batches of size from 45 to 90 depending on a GPU video memory,
- with a learning rate of 0.002,
- and a momentum equal to 0.99.

Training can take up to a week on a server with advanced GPU, such as NVIDIA Titan X or GTX980ti.

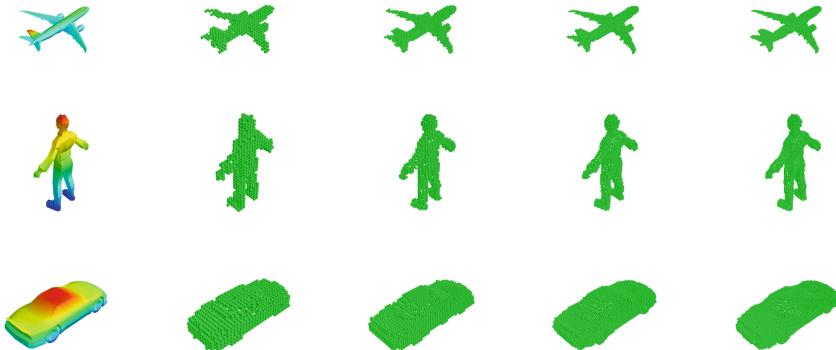


Fig. 2. Examples of some objects voxelizations at different resolutions 30, 50, 70, 100 (from left to right), left-most objects are depicted using original meshes

We train Sparse 3D Convolutional Neural Network (S3DCNN) on the 3D shape classification dataset by splitting it into training and validation subsets, adding augmentation of data to achieve rotational and translational invariance. After training a model on a dataset of pairs, we use it to embed voxel representations of 3D meshes into 192-dimensional space. The retrieval consist of ranking search objects by a cosine distance of vectors from a query vector.

The most popular metrics for evaluating retrieval performance are

- Precision-Recall Curve shows a trade-off between these two measures and how quickly the precision drops with the recall increase,
- Mean average precision (mAP). Given a query, its average precision is the average of all precision values computed on all relevant objects in the retrieved list. Given several queries, the mean average precision (mAP) is the mean of average precisions for these queries.

We evaluated mAP for different voxel rendering sizes of 3D shapes both at train and test times, see also Fig. 2.

To check if our model is comparable with other architectures, we consider a classification task. So, we trained our model for the classification task using the ModelNet40 train subset with

- SoftMax last layer for 200 epochs,
- with exponentially discounting learning rate,
- and performed retrieval evaluation on the test subset,

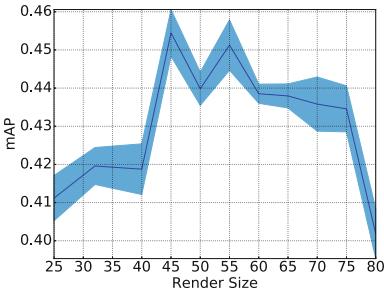


Fig. 3. Dependence of the retrieval performance on the input spatial resolution

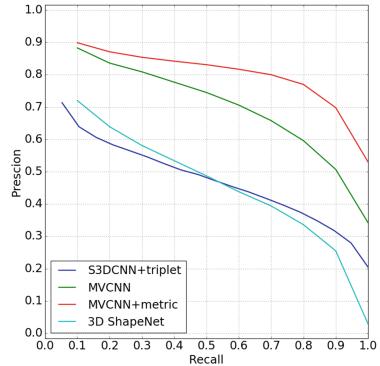


Fig. 4. Precision-Recall curve for our method

- taking 20 images from every class, and ranking them w.r.t their L_2 -norm by activations taken from the 17-th layer.

Results of these experiments are provided in Table 2. We can see that in case of classification task setup our model is comparable in terms of the classification accuracy, but mAP values are worse. But in case of metric learning performance of S3DCNN on mAP metric is much better. Superior performance of retrieval task with MVCNN is not a surprising result, since MVCNN uses neural nets, pre-trained on ImageNet. On the other hand our model only requires 3D Shape dataset to learn.

In Fig. 3 we provide the dependence of mAP on the input spatial resolution. We can see that the retrieval performance improves with increase in the input spatial resolution up to around 45–50, after that it drops slightly and goes to plateau. It can be attributed to the insufficient amount of layers for the same scale of features, that can be separated in higher layers. Light blue color shows range of mAP on validation for top 30 trained architectures.

We would like to note that in Fig. 3 mAP values provided for different validation epochs and variability of best model can be explained by difference in total learning time.

5 Results

We found that the retrieval performance improves with increase in the input spatial resolution. However, such an effect is difficult to check experimentally and to use in practice, as e.g. for usual 3D dense CNNs the computational time is prohibitively large. In our case, data sparsity helps us to process data in reasonable time even with input resolution up to 100^3 voxels, therefore we can benefit from the increase of the input spatial resolution when performing retrieval. In Fig. 4 we can see that our method is comparable to [22] in low

Table 2. Evaluation on Modelnet40

Method	Classification	Retrieval AUC	Retrieval mAP
3DShapeNet [22]	77.32%	49.94%	49.23%
MVCNN [18]	90.10%	—	80.20%
VoxNet [16]	83.00%	—	—
VRN [3]	91.33%	—	—
S3DCNN (proposed)	90.30%	36.05%	33.67%
S3DCNN + triplet (proposed)	—	48.81%	46.71%

recall, and better at higher recall values, that indicates better scalability of our method. In Table 2 for the retrieval we used features from the one before last layer of the network of size 192, which in comparison to 4000 in 3DShapeNet model [22] is 20 times smaller but achieves almost the same retrieval metrics.

We evaluated our network architecture described in Table 1 on popular state-of-the-art frameworks for Deep Learning, such as Tensorflow [1] on GPU and Theano [19] on CPU. Using Keras [5] 2.0.2 with Tensorflow [1] 1.2.1 backend on Nvidia Titan X GPU with 12Gb of GPU memory, we were able to exhaust all of it with batch size equal to 12, and performed forward passes on average 0.0301 s/sample, which is comparable to processing speed of our implementation with render size of about 60–70. Other setup was an implementation of our network architecture on Keras with Theano backend using Intel i7-5820K 6-core CPU processor, took 1.53 s/sample, which is significantly slower.

Acknowledgments. We are very grateful to Dmitry Yarotsky for his contribution to this research project. Big Thanks to Benjamin Graham for some useful comments and ideas. Thanks to Rasim Akhunzyanov for his help in debugging the PySparseConvNet code.

The research was partially supported by the Russian Science Foundation grant (project 14-50-00150). E. Burnaev was partially supported by the Next Generation Skoltech-MIT Program.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). tensorflow.org
2. Bottou, L.: Stochastic gradient tricks. Neural Netw. Tricks Trade Reloaded **7700**, 430–445 (2012)
3. Brock, A., Lim, T., Ritchie, J., Weston, N.: Generative and discriminative voxel modeling with convolutional neural networks. arXiv preprint [arXiv:1608.04236](https://arxiv.org/abs/1608.04236) (2016)
4. Bronstein, A.M., Bronstein, M.M., Guibas, L.J., Ovsjanikov, M.: Shape google: geometric words and expressions for invariant shape retrieval. ACM Trans. Graph. (TOG) **30**(1), 1 (2011)
5. Chollet, F., et al.: Keras (2015). <https://github.com/fchollet/keras>

6. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 248–255. IEEE (2009)
7. Graham, B.: Spatially-sparse convolutional neural networks. arXiv preprint [arXiv:1409.6070](https://arxiv.org/abs/1409.6070) (2014)
8. Hegde, V., Zadeh, R.: Fusionnet: 3d object classification using multiple data representations. arXiv preprint [arXiv:1607.05695](https://arxiv.org/abs/1607.05695) (2016)
9. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: Feragen, A., Pelillo, M., Loog, M. (eds.) SIMBAD 2015. LNCS, vol. 9370, pp. 84–92. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24261-3_7
10. Johns, E., Leutenegger, S., Davison, A.J.: Pairwise decomposition of image sequences for active multi-view recognition. arXiv preprint [arXiv:1605.08359](https://arxiv.org/abs/1605.08359) (2016)
11. Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Rotation invariant spherical harmonic representation of 3D shape descriptors. Symp. Geom. Process. **6**, 156–164 (2003)
12. Knopp, J., Prasad, M., Willem, G., Timofte, R., Van Gool, L.: Hough transform and 3D SURF for robust three dimensional classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6316, pp. 589–602. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15567-3_43
13. Kokkinos, I., Bronstein, M.M., Litman, R., Bronstein, A.M.: Intrinsic shape context descriptors for deformable shapes. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 159–166. IEEE (2012)
14. LeCun, Y., Bengio, Y.: Convolutional networks for images, speech, and time series. In: The Handbook of Brain Theory and Neural Networks, vol. 3361, no. 10 (1995)
15. Mathieu, M., Henaff, M., LeCun, Y.: Fast training of convolutional networks through ffts. arXiv preprint [arXiv:1312.5851](https://arxiv.org/abs/1312.5851) (2013)
16. Maturana, D., Scherer, S.: Voxnet: a 3d convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 922–928. IEEE (2015)
17. Sedaghat, N., Zolfaghari, M., Brox, T.: Orientation-boosted voxel nets for 3D object recognition. arXiv preprint [arXiv:1604.03351](https://arxiv.org/abs/1604.03351) (2016)
18. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.G.: Multi-view convolutional neural networks for 3D shape recognition. In: Proceedings of ICCV (2015)
19. Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints, abs/1605.02688, May 2016
20. Tokui, S., Oono, K., Hido, S., Clayton, J.: Chainer: a next-generation open source framework for deep learning. In: Proceedings of Workshop on Machine Learning Systems (LearningSys) in the Twenty-Ninth Annual Conference on Neural Information Processing Systems (NIPS) (2015)
21. Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y.: Learning fine-grained image similarity with deep ranking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1386–1393 (2014)
22. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3D shapenets: a deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1912–1920 (2015)

Floor-Ladder Framework for Human Face Beautification

Yulia Novskaya^(✉), Sun Ruoqi, Hengliang Zhu, and Lizhuang Ma

School of Electronic Information and Electrical Engineering,

Shanghai Jiao Tong University, Shanghai, China

{yulia_novskaya,ruosunqi}@sjtu.edu.cn, hengliang_zhu@163.com,
ma-lz@cs.sjtu.edu.cn

Abstract. In this paper, we propose a Floor-Ladder Framework (FLN) based on age evolution rules to generate beautified human faces. Beside the shape of faces, younger faces achieve more attractiveness. Thus we process the beautiful face by applying the reversed aging rules. Inspired by the layered optimization methods, the FLN adopts three floors and each floor contains two ladders: the Single Layer Older Neural Network (SLONN) and the extended Skull Model. The Peak Shift algorithm is designed to train the SLONN aiming to capture the reversed aging rules of the face skin. Due to the growth rules of the face shape, we extended the Skull Model by adding Marquardt Mask. Given the input portrait, our algorithm effectively produces a beautified human face without losing personal features.

Keywords: Face beautification · Floor-Ladder Framework

The Peak Shift Algorithm

The Single Layer Older Neural Network · The extended Skull Model

1 Introduction

Face beautification has a wide range of applications in the daily life. Individuals who look neat and clean will not only show respect and please to others, but also enhance their confidence and inner strength [4]. Thus, there exists a variant of methods to improve the beauty of faces, including makeup, plastic surgery, etc. Two key issues of face aging process are the face texture and the face shape. Firstly, we improve the face skin beautification method. In terms of the variance of age features and the invariance of personalized features, algorithms aiming to divide faces into special layers with different functions have gained more attention. A variant of face beautification algorithms [7, 20–22] utilize the detail or the age layer to keep the nature of human face. However, those layered methods are designed specifically for these algorithms which can hardly meet our requirements. In this paper, we propose a novel method based on Gaussian model to extract the detail of faces, which effectively eliminate the overexposure problem. Along with the process of aging, there will be some defects appeared on the

face, such as wrinkles, splashes etc. In terms of assumption, Chen et al. [7] proposed a face beautification method based on the reversed age processing. They apply SVM to learn the aging features on the detail layer of faces, which shows younger faces have fewer defects than the aged ones. This method produces natural results, but it focuses on the global features, while pays little attention to the local characteristics. Face aging is a local driven process, which means that the face growing older started from a wrinkle or a spot.

Deep learning algorithms perform well in solving high-dimension problems for nonlinear mapping, which contains several popular networks: the deep neural network [5, 11, 12, 33], the deep Boltzmann machine [13, 14], the deep Bayesian network [6, 15, 16]. The nonlinear features extracted by neural networks are effective in many tasks. Thus, combining the short term feature selection method [1] and the feature separation training method [6], we propose the Floor-Ladder Framework to simulate the reversed aging revolution process. Figure 1 represents introduced framework for face beautification. The Floor-Ladder Framework has three floors and each floor has two ladders. Floors represent different generation groups, while the ladders contain the process of modifying facial texture and shape. We divide all the faces into three groups according to the age range: 0–20, 21–35 and 36–80. Then we can learn the special rules which can adapt to different generations. We also implement the SLONN reflecting the aging rule of face defects.

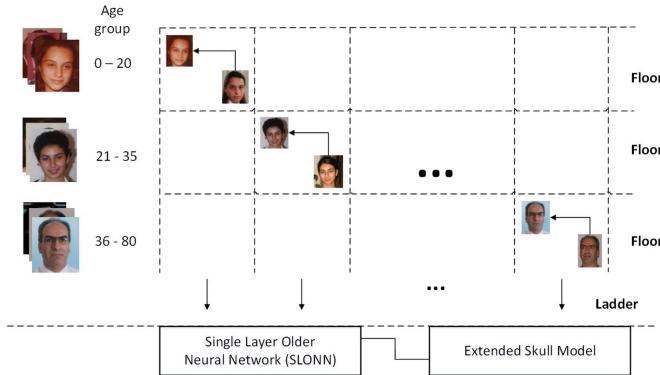


Fig. 1. The Floor-Ladder Framework contains three floors and two ladders. The floor represents the age group, while the ladder represents the beautification algorithm for each age group

Face shape beautification is also an important way to improve the attractiveness of human faces in the image. Todd et al. [9] proposes the facial growth model which shows that the face shape changes with age. They perform a hydrostatic analysis based on the gravity to generate the growth rule of human faces. One of the models [9] is expressed in Fig. 2. The human faces become larger and longer when a man grows up in the result of the pressure. Thus, we apply the

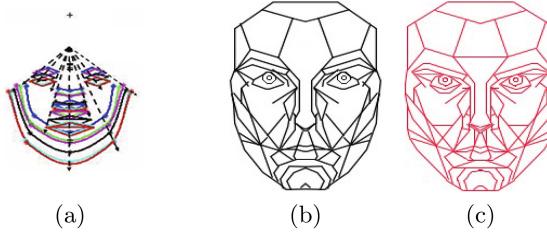


Fig. 2. (a) The extended skull model. The Maquardt mask made for women (b), while another is made for man (c).

growth rule of the model to the input face, and generate a face with the younger shape.

The Skull model provides a great visual impact and it can only represent the growth rule of face shape from 0 years old to 20 years old. It is necessary to learn the growth rule when the person is older than 20. The Marquardt mask is an ideal beautiful face model, which is adapted to all the races. Generally, we hold that the human faces which are closer to the mask are more beautiful. The authors [30] show that human faces go away from the Marquardt mask, as the growth of age. Thus, we extend the skull model by adding the Marquardt mask to generate adult faces.

In particular, we combine these methods and achieve efficient improvements in the experimental results. The contribution of this paper is shown as follows. Firstly, we propose the Floor-Ladder Framework to generate beautiful human faces, which contains totally five stages and is trained by the Peak Shift algorithm. Also, we update weights in SLONN by adding the Gaussian Distribution to get the continuous result and finally in the fifth stage we combined skull model and the Marquardt Mask to gain the extended Skull Model.

This paper is organized as follows: in Sect. 2, we overview the proposed approach of the face beautification and face aging. In Sect. 3, we describe the Floor-Ladder Framework in detail. In Sect. 4, we show our experiments with comparisons. Finally, we conclude this paper and discuss the future work.

2 Related Work

In recent years, a considerable number of researchers are interested in the face beautification methods. These methods are brought up into two ways: the directed face beautification and the face makeup. In this paper, we beautify human faces by generating younger face in both shape and skin based on reversed age revolution, which produces more natural results.

2.1 Directed Face Beautification

The wide application of machine learning methods especially DCNNs access a huge success in the image processing area, including identification, recognition,

reconstruction. Face beautification also benefits from the method and makes a rapid development. Two directions, one as face texture and other as face shape, have recently demonstrated remarkable results for beautification.

Face skin beautification is designed to reduce the wrinkles and spots to increase the luster of the skin. The most basic way to achieve the effect is to use a low-pass filter. However, the filter destroys the personal information on the face while it removes the wrinkles. To protect the nature and reality of the human face, the layered model is proposed to protect the user from changing the basic skin texture. Chen et al. [7] proposes the layered method based on $L^*a^*b^*$ color space because this color space can divide the face skin into structure and color parts. Then they can apply the reversed face aging method to beautify human faces without changing personal information. In this paper, we improve Chen's method [7] to better adapt to the Floor-Ladder Framework.

Face shape beautification method is proposed to morph the input human face to a more attractive one. Zhang et al. [17] transfers human faces toward and away from the average face shapes to prove that the human faces which are similar to the average ones look more beautiful. Leyvad et al. [18] utilizes Support Vector Regression (SVR) to learn the beauty score and morphs the input face to the one with higher beauty score. These methods lead to great results. Li et al. [19] improves the score-learning method by using deep learning instead of SVR, which generally produce more natural results. According to the requirement of being natural, we generate the human face by using the improved skull model based on the face growth rule.

2.2 Face Aging Methods

The researchers of face aging achieve great success by using the machine learning methods based on a large amount of human faces. Face aging is of fundamental importance to various domains, including cross-age face verification and recognition. Kemelmachershizerman et al. [2] applies the average face skin and skull model, which can produce the aging face from 1 to 80 years old. This method can adapt to various illuminations. Shu et al. [1] proposes an aging dictionary to modeling face aging based on the neighbor age groups. Wang et al. [23] proposed a recurrent face aging framework based on RNN. An increasing number of the face aging methods [1, 2, 23, 24] provide improvement based on these two characteristics. In this paper, we also take them into consideration and provide the special improvements of skull model and texture optimization algorithm in our reversed face aging methods.

2.3 Face Makeup Methods

Face makeup is the technology to improve the appearance of human faces, which is widely used by women. We can improve the attractiveness of a human by choosing a proper haircut and adaptive makeup. In 2016, Lee et al. [22] improves the makeup transfer method by using the Gaussian weight map. Liu et al. [27] proposes a deep face makeup transfer method to synthesise the makeup on the

female face. The Euclidean method is used to choose the suitable makeup faces based on the deep feature, while the deep localized makeup transfer network synthesizes the beautified face. The beautified face has various makeup lightness ranging from light to dark. These face makeup methods are well designed for the female users.

2.4 Commercial Systems

Many commercial systems are used to generate beautified faces. Portraiture [29] is a plug-in of the Photoshop with a masking tool that enables selective smoothing face's skin. MeituPic [28] is an image editing software, which offers various filters for face beautification. We demonstrate a comparison between these commercial systems and the proposed method.

3 Floor-Ladder Framework

Floor-Ladder Framework is a nonlinear framework based on deep neural network, which processes human face from neighboring generations, which contains three floors and two ladders. The floors in the framework represent different age groups, while the ladders are the methods to generate younger faces. The first ladder is the SLONN, while the second ladder is the extended Skull Model.

We train the SLONN to generate younger texture for different age groups without losing the special features of the corresponding generation. The aging parameters are features learned from the details of layers that is extracted from the human skin. Then we use tested faces as the input for the SLONN to generate the younger face skin. The face shape changes as the person grow up. So, we extend the Skull Model [9] by adding the Marquardt mask. The Skull Model only records the face shape growth rule from age 0 to 20. The face shape goes away from the Marquardt mask when the person is older than 20. Thus we combine them to represent the face shape growth rule and apply them to the input faces to process the younger face shape.

The workflow of proposed framework is illustrated in Fig. 3. There are 3 main steps. Firstly, we decompose both training and test data into three layers by decomposition. Secondly, we train weights and bias of the network on the detail layer separately. Thirdly, we combine the reconstructed detail layer, lighting and color layers, to obtain a complete image. Finally, we apply the extended skull model depending on the age and gender of the input image.

3.1 Training Dataset and Test Data

In this paper, some face images are obtained from the FG-Net database [3] and others are collected from the MORPH dataset [32]. Furthermore, we drop out one channel images and blurred images in the datasets to constitute the required dataset. We used around 1800 pictures of our database as the training set and around 180 pictures as the validation set, around 180 pictures of our dataset as

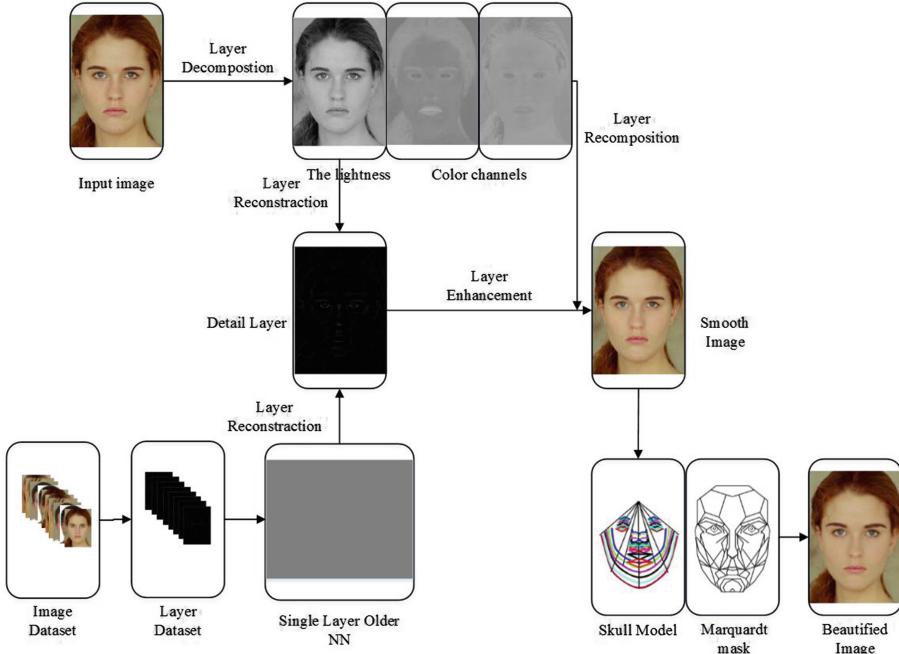


Fig. 3. The workflow of our approach. Step 1: Image layer decomposition. Step 2: Single Layer Older NN learning. Step 3: Extended Skull Model by Marquardt mask.

a test set. All images were resized to 100×150 pixels. We extract the face skin texture by using the face detection method [8]. The Explicit Shape Regression (ESR) algorithm [31] was used for face alignment. We generate 88 points on the human faces based on this algorithm, which contains the shape of faces and the five sense organs. To protect the personality, we extract the age layer and the identity layer by using the extended weighted-last-squares (WLS) method [10]. Nouveaeu-Richard and Lacharrie're [26] researchers established the relationship of different face areas and the growth speed of wrinkles. The growth speeds of wrinkles are different in different areas. Thus, we establish the SLONN to learn the growth rule in local areas.

3.2 Single Layer Older Neural Network (SLONN)

We use the SLONN to process the texture of the human face on the first ladder of every floor. The weights and biases are trained based on extracted detail layers.

Training Weights. We use the Peak Shift method to train the weights of the SLONN based on the pixel's order. The peak shift means that we shift the next Gaussian peak of the SLONN. When the loss is positive, the next Gaussian peak

is the pixel on the right of the current peak, otherwise, we choose the left one. The delta is calculated by using the formula:

$$\Delta = I^y - h(x) \quad (1)$$

where the I^y is the ground truth, the x is the input image and the $h(x)$ is the output image. The output image for train the network is next existed image in the dataset, age-target older than the initial one. The delta represents the difference of ground truth and output value, which is used to update weights in SLONN. We train weights by using Gaussian Distribution:

$$p_{ijk} = p_{ijk} + |\Delta|G_k \quad (2)$$

$$G_k = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(k-\mu)^2}{2\sigma^2}} \quad (3)$$

where i, j , represent the position of the center pixel of the grid. k is the order of pixels. G denotes the Gaussian distribution model. μ is the mean and σ is the variance. In this paper, we set $\mu = 0$ and $\sigma = 0.5$. The back propagation algorithm is a basic method to train the network. So, we calculate the loss and update the weights in the network, which is used to train the network effectively. The value of image pixel is continuous in the local area and the weight is deal with a pixel in neighbor location, which means that the values of two weights are close to each other. So, we add the Gaussian distribution instead of adding the delta to the specific pixel to get a continuous transformation. Finally, we normalized output data of the SLONN to keep images in the same color range with the input images.

Training Bias. The same person has similar skin texture in his or her whole life compared to another person. With the time going by, the texture is changing according to the environment. We learned the age-related parameters based on the standard deviation in each generation by using support vector machine (SVM). First, we calculate the standard deviation σ of the detail layers:

$$\sigma^2 = \sum_i (p_i - \mu)^2 \quad (4)$$

Then we update each value by reducing the whole standard deviation:

$$I_{ij} = \mu + \frac{\sigma_{age-pre}}{\sigma_{age-img}} (I_{ij} - \mu) \quad (5)$$

We extract the detail layer which does not contain the color so that this algorithm can benefit all the ethnic groups.

3.3 Extended Skull Model

The Skull Model [9] shows the growth rules of human faces younger than twenty. With the person growing up, the face shape goes away from the Marquart mask.

According to the findings, the Skull Model is extended by adding the Marquardt Mask which is shown in Fig. 2. After that, we use two masks to generate the face shape for male and female respectively. By considering the difference between two parts of the model, we develop two methods to change the face shape. The first one is an image-based algorithm. We use this method to search for the most similar face shape in the same age group of the input image. Furthermore, we assume that the similar input faces have similar younger face shapes. Thus, we morph the input face to the corresponding younger faces shape. However, this issue is not entirely straightforward and we cannot find all the corresponding younger faces in some case. The second method named model-based method which is used to morph the face to the Marquardt mask by using the warping method [7]. We also add the features of symmetrical faces to the beautification process to enhance the attractiveness.

4 Experiments and Results

For the execution of the experiments, we used Matlab R2016b. In term of computation, the approximate training time is 1.5 min. The proposed method divides the human face into 3 age groups based on the age ranges: 0–20, 21–35, 36–80. Normalizing can reduce the damage which is caused by different image numbers.



Fig. 4. In comparison to the face texture beautification algorithms. (a) Original image. (b) Portaiture. (c) Liang et al. (d) MeituPic. (e) Our result.

The growth rhythm of wrinkles is different in each generation. As is known to all of us, the face with fewer wrinkles or stains is considered to be healthier and more beautiful, which accelerates the development of facial texture beautification techniques. In comparison to previous face texture beautification methods proposed by Liang et al. [25], our algorithm generates more natural face skin without strange color, but it cannot deal with all the defects on the faces. The result is shown in Fig. 4. Furthermore, we compare our result with two commercial systems. We apply chosen commercial systems on the same test images, and use the more appropriate technique for face beautification. As a plug-in in Photoshop CS 6, we run Portraiture with proposed default settings for face image enhancement. For MeituPic, we use one of four main functions to beautify face image. All these methods accomplish good results in face beautification, but there were obvious differences among their results. MeituPic performed an unremarkably smooth operation on the input image. Moreover, MeituPic globally applies some filters to the whole image, without detecting face and skin region of the image. Portraiture successfully accomplishes significantly satisfying result in removing unwanted spots on the face. Figure 5 shows beautification results of proposed design in comparison with other methods. Leyvand et al. [18] morph



Fig. 5. The beautification result with a comparison to the face shows beautification algorithm. The images in the first row are original portraits. The algorithm proposed by Leyvand et al. [18]. Images in the last row are result of the proposed algorithm.

human faces towards the face with higher beauty score predicted by SVR, while we use the extended Skull Model.

We print these faces in the form of color printing and find 30 volunteers in the campus. They are forced to vote to more beautiful faces between previous results and our results. In the face texture exam, 12 raters like our results for the natural appearances, while 18 raters prefer the smoother face skin. In the face shape exam, we found that 25 raters (more than 80%) prefer our beautification results. Generally, they hold that produced results are younger and more beautiful faces than input faces. We found that there are several limitations in our method. Our framework focuses on the front face, which can only deal with the front human faces for that SLONN is set to train the texture in the corresponding position. We are aiming to change the structure of the SLONN to establish a nonlinear mapping to apply the transformation to the large-scale position of human faces. It seems advisable that in the future we could extend this framework to generate the younger face sequence of different generations and different poses of the face.

5 Conclusion

In this paper, we described an efficient algorithm for the beautification of the face using Floor-Ladder Framework to automatically generate beautified faces with generation rules to reverse the human faces' age. This framework contains 2 main steps: the SLONN and the extended Skull Model. Our method can benefit all the races and it can adapt to both male and female faces. The extended Skull Model is adapted to all the human beings. The global texture transformation rule contains the detail layers of all ethnic groups. The comparison with some recent existing algorithms and commercial systems shows that the proposed framework has strong ability to process a younger face in the neighbor generation. Our method produces the beautiful face without losing the personalities of the original faces and achieves higher scores than previous methods.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (No.61472245), and the Science and Technology Commission of Shanghai Municipality Program (No.16511101300).

References

1. Shu, X., Tang, J., Lai, H., et al.: Personalized age progression with aging dictionary. In: IEEE International Conference on Computer Vision (ICCV), pp. 3970–3978 (2015)
2. Kemelmachershizerman, I., Suwajanakorn, S., Seitz, S.M.: Illumination-aware age progression. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3334–3341 (2014)
3. FG-NET Aging Database. <http://sting.cyclege.ac.cy/alanitis/fnetaging/>
4. Alley, T.R.: Social and Applied Aspects of Perceiving Faces. Lawrence Erlbaum Associates, Hillsdale (1988)

5. Yang, J., Price, B., Cohen, S., et al.: Object contour detection with a fully convolutional encoder-decoder network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
6. Chi, N.D., Luu, K., Quach, K.G., et al.: Beyond principal components: deep Boltzmann machines for face modeling. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4786–4794 (2015)
7. Chen, Y., Ding, S.H., Gan-Le, H.U., et al.: Facial beautification method based on age evolution. *Comput. Aided Drafting Des. Manufact.* **4**, 7–13 (2013)
8. Viola, P., Jones, M.J.: Robust real-time face detection. *Int. J. Comput. Vis.* **57**(2), 137–154 (2004)
9. Todd, J.T., Mark, L.S., Shaw, R.E., et al.: The perception of human growth. *Sci. Am.* **242**(2), 132–144 (1980)
10. Lanitis, A., Taylors, C.J., Cootes, T.F.: Toward automatic simulation of aging effects on face images. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(24), 442–455 (2002)
11. Sun, Y., Chen, Y., Wan, X., Tang, X.: Deep learning face representation by joint identification-verification. In: Advances in Neural Information Processing Systems (NIPS), pp. 1988–1996 (2014)
12. Yu, L., Michael, S.L.: Learning relaxed deep supervision for better edge detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
13. Chi, N.D., Khoa, L., Kha, G.Q., et al.: Longitudinal face modeling via temporal deep restricted Boltzmann machines. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
14. Timo, B., Stefanie, W.: A groupwise multilinear correspondence optimization for 3D faces. In: The IEEE International Conference on Computer Vision (ICCV), pp. 3604–3612 (2015)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
16. Zhang, Y., et al.: Improving object detection with deep convolutional networks via Bayesian optimization and structured prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 249–258 (2015)
17. Zhang, D., Zhao, Q., Chen, F.: Quantitative analysis of human facial beauty using geometric features. *Pattern Recogn.* **44**(4), 940–950 (2011)
18. Leyvand, T., Cohen, D., Dror, G., et al.: Data-driven enhancement of facial attractiveness. *ACM Trans. Graph. (TOG)* **27**(3), 15–19 (2008)
19. Li, J., Xiong, C., Liu, L., et al.: Deep face beautification. In: ACM International Conference on Multimedia, pp. 793–794 (2015)
20. Guo, D., Sim, T.: Digital face makeup by example. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
21. Liu, L., Xu, H., Xing, S., et al.: Wow! You are so beautiful today! In: Proceedings of 21st ACM International Conference on Multimedia (MM), vol. 11, no. 1s, pp. 3–12 (2014)
22. Lee, J.Y., Kang, H.B.A.: New digital face makeup method. In: IEEE International Conference on Consumer Electronics (2016)
23. Wang, W., Cui, Z., Yan, Y., et al.: Recurrent face aging. In: IEEE Conference on Computer Vision and Pattern Recognition(CVPR) (2016)
24. Sethuram, A., Ricanel, K., Patterson, E.: A hierarchical approach to facial aging. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 100–107 (2010)
25. Liang, L., Jin, L., Li, X.: Facial skin beautification using adaptive region-aware masks. *IEEE Trans. Cybern.* **44**(12), 2600–2612 (2014)

26. Nouveaeu-Richard, S., de Lacharrie're, O.: Skin aging: an exploratory research above the Chinese and European. In: China Cosmetic Academic Symposium (2006)
27. Liu, S., Ou, X., Qian, R., et al.: Makeup like a superstar: Deep Localized Makeup Transfer Network (2016)
28. Meitulnc, Meitupic (2014). <http://xiuxiu.meitu.com/en/>
29. Lightroom Imagenomic, "Portraiture" (2015). <http://Imagenomic.com/pt.aspx>
30. Face variations by age. In: Network: Marquardt Beauty Analysis. <http://www.beautyanalysis.com/beauty-and-you/face-variations-age/>
31. Cao, X., Wei, Y., Wen, F., et al.: Face alignment by explicit shape regression. In: CVPR (2016)
32. MORPH Database. <http://www.faceaginggroup.com/morph/>
33. Zhu, S., Liu, S., Loy, C.C., Tang, X., et al.: Cascaded bi-network for face hallucination. In: ECCV (2016)

Array DBMS and Satellite Imagery: Towards Big Raster Data in the Cloud

Ramon Antonio Rodriges Zalipynis^(✉), Evgeniy Pozdeev, and Anton Bryukhov

National Research University Higher School of Economics, Moscow, Russia
rodrigues@wikience.org, jonnypozdeev@gmail.com, asbryukhov@gmail.com

Abstract. Satellite imagery have always been “big” data. Array DBMS is one of the tools to streamline raster data processing. However, raster data are usually stored in files, not in databases. Respective command line tools have long been developed to process these files. Most of the tools are feature-rich and free but optimized for a single machine. The approach of partially delegating *in situ* raster data processing to such tools has been recently proposed. The approach includes a new formal N -d array data model to abstract from the files and the tools as well as new formal distributed algorithms based on the model. ChronosServer is a distributed array DBMS under development into which the approach is being integrated. This paper extends the approach with a new algorithm for the reshaping (tiling) of arbitrary N -d arrays onto a set of overlapping N -d arrays with a fixed shape. Cutting arrays with an overlap enables to perform a broad range of large imagery processing operations in a distributed shared-nothing fashion. Currently ChronosServer provides a rich collection of raster operations at scale and outperforms SciDB up to $80\times$ on Landsat data. SciDB is the only freely available distributed array DBMS to date. Experiments were carried out on 8- and 16-node clusters in Microsoft Azure Cloud.

Keywords: ChronosServer · SciDB · Cloud computing · Array DBMS · Satellite imagery · *In situ* · Command line tools · Big data · Landsat

1 Introduction

Satellite sector is data-rich, practically important and commercially attractive. For example, the Landsat Program is the longest continuous space-based record of Earth’s land in existence running from 1972 onwards. It has accumulated over 6.8×10^6 scenes mostly in GeoTIFF format (one scene is ≈ 1 GB) [13]. Landsat popularity made Amazon and Google to provide Landsat scenes via commercial clouds [7]. The number of practical Landsat applications is rapidly growing [12].

File-based raster data storage resulted in a broad set of highly optimized raster file formats. For example, GeoTIFF is an effort by over 160 companies and organizations to establish interchange format for georeferenced raster

imagery [8]. Decades of development resulted in many elaborate tools for processing these files: work on ImageMagic started in 1987 [10], GDAL (Geospatial Data Abstraction Library) has $\approx 10^6$ lines of code and hundreds of contributors [5].

The idea of partially delegating raster data processing to existing command line tools was first presented and proved to outperform SciDB on NetCDF data $3\times$ to $193\times$ on a single machine [21] and $1000\times$ running both SciDB and ChronosServer on a computer cluster (Microsoft Azure Cloud) [22]. ChronosServer is the system into which the delegation ability is being integrated [20].

The formal array model and formal distributed algorithms are given in [22]. The new two-level data model was designed to uniformly represent diverse raster data types and formats, take into account the distributed context, and be independent of the underlying raster file formats at the same time [22].

This paper further shows that existing tools can equip an array DBMS with a powerful and rich functionality. Many raster processing algorithms require significant implementation efforts but already existing command line tools are largely ignored in the array DBMS research field. SciDB was first released in 2008 and still lacks (July 2017) even core raster operations like interpolation [11].

Satellite imagery arrive at ChronosServer in their native file formats. Then, files are preprocessed (Sect. 3) which is much faster than SciDB import (Sect. 5). The storage layer of ChronosServer sits on top of files in diverse raster formats distributed between cluster nodes. For the sake of completeness, Sect. 2 formally describes ChronosServer data model [22] which is basic for distributed raster processing algorithms in Sect. 4. Algorithms delegate portions of work to GDAL and ImageMagic within a single cluster node by launching them directly on a file. ChronosServer is responsible for proper data exchange between cluster nodes. This synergy of existing raster file formats, command line tools, and their distributed orchestration outperforms SciDB from $5\times$ to $80\times$ (Sect. 5).

2 ChronosServer

2.1 ChronosServer Multidimensional Array Model

In this paper, an N -dimensional array (N -d array) is the mapping $A : D_1 \times D_2 \times \dots \times D_N \mapsto \mathbb{T}$, where $N > 0$, $D_i = [0, l_i) \subset \mathbb{Z}$, $0 < l_i$ is a finite integer which is said to be the *size* or *length* of i th dimension, and \mathbb{T} is a standard numeric type. In this paper, $i \in [1, N] \subset \mathbb{Z}$. Let us denote the N -d array A by

$$A\langle l_1, l_2, \dots, l_N \rangle : \mathbb{T} \quad (1)$$

By $l_1 \times l_2 \times \dots \times l_N$ denote the *shape* of A , by $|A|$ denote the *size* of A such that $|A| = \prod_i l_i$. A *cell* value of A with index (x_1, x_2, \dots, x_N) is referred to as $A[x_1, x_2, \dots, x_N]$, where $x_i \in D_i$. Each cell value of A is of type \mathbb{T} . An array may be initialized after its definition by listing its cell values: $A\langle 2, 2 \rangle : \text{int} = \{\{1, 2\}, \{3, 4\}\}$, where A is 2-d array of integers, $A[0, 1] = 3$, $|A| = 4$, and the shape of A is 2×2 .

Indexes x_i are optionally mapped to specific values of i th dimension by *coordinate* arrays $A.d_i\langle l_i \rangle : \mathbb{T}_i$, where \mathbb{T}_i is a totally ordered set, and $d_i[j] < d_i[j+1]$ for all $j \in D_i$. In this case, A is defined as

$$A(d_1, d_2, \dots, d_N) : \mathbb{T} \quad (2)$$

A *hyperslab* $A' \sqsubseteq A$ is an N -d subarray of A defined by the notation

$$A[b_1 : e_1, \dots, b_N : e_N] = A'(d'_1, \dots, d'_N) \quad (3)$$

where $b_i, e_i \in \mathbb{Z}$, $0 \leq b_i \leq e_i < l_i$, $d'_i = d_i[b_i : e_i]$, $|d'_i| = e_i - b_i + 1$, and for all $y_i \in [0, e_i - b_i]$ the following holds

$$A'[y_1, \dots, y_N] = A[y_1 + b_1, \dots, y_N + b_N] \quad (4a)$$

$$d'_i[y_i] = d_i[y_i + b_i] \quad (4b)$$

Equations (4a) and (4b) state that A and A' have a common coordinate subspace over which cell values of A and A' coincide. Note that the original dimensionality is preserved even if some $b_i = e_i$ (in this case, “ $: e_i$ ” may be omitted in (3)). Also, “ $b_i : e_i$ ” may be omitted in (3) if $b_i = 0$ and $e_i = |d_i| - 1$.

2.2 ChronosServer Datasets

A *dataset* $\mathbb{D} = (A, M, P)$ contains a *user-level* array $A(d_1, \dots, d_N) : \mathbb{T}$ and the set of *system-level* arrays $P = \{(A_k, B_k, E_k, M_k, K_k, node_k)\}$, where $A_k \sqsubseteq A$, $k \in \mathbb{N}$, $node_k$ is the cluster node storing A_k , M_k is metadata for A_k , $B\langle N \rangle : \text{int} = \{b_1, \dots, b_N\}$, $E\langle N \rangle : \text{int} = \{e_1, \dots, e_N\}$ such that $A_k = A[b_1 : e_1, \dots, b_N : e_N]$, and K_k is the key of A_k assigned on the data preprocessing stage (Sect. 3). A user-level array is never materialized and stored explicitly: operations with A are mapped to operations with respective arrays A_k . Let us call a user-level array and a system-level array an array and a subarray respectively for short. Dataset metadata $M = \{(key, val)\}$ includes general dataset properties (name, description, contacts, etc.) and metadata valid for all $p \in P$ (data type \mathbb{T} , storage format, date, cartographic projection, etc.). For example, $M = \{(name = "Landsat 8 Band 1"), (type = \text{int16}), (format = \text{GeoTIFF})\}$. Let us refer to an element in a tuple $p = (A_k, B_k, \dots) \in P$ as $p.A$ for A_k , $p.B$ for B_k , etc.

2.3 ChronosServer Architecture

ChronosServer runs on a computer cluster of commodity hardware. Files of diverse raster file formats are distributed between cluster nodes without changing their formats. A file is always stored entirely on a node in contrast to parallel or distributed file systems. Workers are launched at each node and are responsible for data processing. A single Gate at a dedicated node receives client queries and coordinates workers.

Gate stores metadata for all datasets and subarrays. Consider a dataset $\mathbb{D} = (A, M, P)$. Arrays $A.d_i$ and elements of $\forall p \in P$ except $p.A$ are stored on Gate. In practice, array axes usually have coordinates such that $A.d_i[j] = start + j \times step$, where $j \in [0, |A.d_i|) \subset \mathbb{N}$, $start, step \in \mathbb{R}$. Only $|A.d_i|$, $start$ and $step$ values have to be usually stored. ChronosServer array model merit is that it has been designed to be generic as much as possible but allowing to establish 1:1 mapping of a $p \in P$ to a real dataset file at the same time.

Upon startup workers connect to Gate and receive the list of all available datasets and file naming rules. Workers scan their local file systems to discover datasets and create $p.M$, $p.B$, $p.E$ by parsing file names or reading file metadata. Workers transmit to Gate the described information.

3 Generic N -d Retiling Algorithm with an Overlap

Data providers disseminate satellite imagery in very diverse forms. For example, Landsat scenes overlap irregularly and sides of a scene bounding polygon are not parallel to the coordinate axes. A satellite usually does not capture exactly the same area during each pass due to the orbit drift: scenes for different dates may overlap for 99% and be shifted relatively to each other. Even basic raster operations with raw scenes from the same dataset are complex.

Algorithm 1 performs tiling of subarrays from their initial diverse forms to a fixed one: N -d arrays with shape $s_1 \times s_2 \times \dots \times s_N$ and edges parallel to the coordinate axes. The algorithm is used on the “data cooking” (data preprocessing) stage. Uniformity of subarrays greatly simplifies raster algorithms (Sect. 4). More importantly, often this kind of data transformation makes it possible to leverage existing command line tools. For example, some tools refuse to perform operations on rasters that do not cover exactly the same area. The data cooking time is negligible compared to SciDB data import time (Sect. 5).

The basic idea is to cut each $p \in P$ onto smaller pieces $P' = \{p' : p' \sqsubseteq p\}$, assign each piece a key, and merge all pieces with the same key into a single, new subarray. For $x \in \mathbb{Z}$, $lag(x) = 0$ if $x \leq 0$, $lag(x) = x - 1$ if $x \geq 1$. We skip detailed description of the algorithm due to space constraints.

Thick blue lines on Fig. 1 are borders of the initial subarrays, dashed red lines are borders of the resulting subarrays (an overlap is not shown). Resulting subarrays will share border cells if overlap is given: e.g., if $\rho_i = 1$, $A[5:8, 1:4]$ and $A[5:8, 3:6]$ will share $A[5:8, 3:4]$. This allows to perform many raster operations in parallel without network exchange of border cell values (e.g. calculate convolution for cells $A[2:6, 4:4]$, Sect. 4.1).

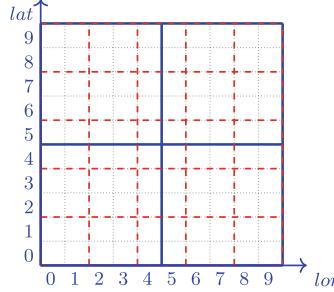


Fig. 1. An example of system-level arrays retiling: $s_1 = s_2 = 2$. (Color figure online)

Algorithm 1. Generic N-d Retiling with an Overlap.

Input: $\mathbb{D} = (A, M, P)$ \triangleright Dataset, section 2.2
 $S = (s_1, s_2, \dots, s_N)$ \triangleright Target shape for subarrays from P
 $\rho = (\rho_1, \rho_2, \dots, \rho_N)$ \triangleright Overlap

Output: $\mathbb{D}' = (A, M, P')$ \triangleright A is the same, $\forall p' \in P'$ shape is $s'_1 \times s'_2 \times \dots \times s'_N$,
 \triangleright where $s'_i = s_i + \rho_i \times 2$ (except border cases)

Require: $s_i \in [1, \Theta_{axis}] \subset \mathbb{N}$, $\prod_{i=1}^N s_i \leq \Theta_{shape}$, $0 \leq \rho_i < s_i \text{ div } 2$ \triangleright Resulting pieces are not too large

1: **function** RETILE(\mathbb{D}, S, ρ)

2: $C \leftarrow \{\}$ and $\mathbb{K} \leftarrow \{\}$ \triangleright C : line 12, \mathbb{K} : line 13 of **procedure** CUT-ONE

3: **for each** $p \in P$ **do** CUT-ONE($C, \mathbb{K}, p, S, \rho$)

4: **for each** $key \in \mathbb{K}$ **do**

5: $C \leftarrow \{a \in C : a.key = key\}$

6: $p_{new} \leftarrow \text{merge all } a \in C \text{ given } a.B_{new} \text{ and } a.E_{new}$ \triangleright DELEGATION

7: $P' \leftarrow P' \cup \{p_{new}\}$

8: **return** $\mathbb{D}' = (A, M, P')$

1: **procedure** CUT-ONE($C, \mathbb{K}, p, S, \rho$)

2: $x_i^b \leftarrow b_i \text{ div } s_i$ $\triangleright b_i = p.B[i], e_i = p.E[i]$

3: $x_i^e \leftarrow e_i \text{ div } s_i$

4: **for each** $y_i \in [x_i^b - \text{sgn}(x_i^b), x_i^e + \text{sgn}((|A.d_i| - 1) \text{ div } s_i - x_i^e)] \subset \mathbb{Z}$ **do**

5: $b'_i \leftarrow \max(y_i \times s_i - b_i - \rho_i, 0)$

6: $e'_i \leftarrow \min((y_i + 1) \times s_i - b_i + \rho_i, e_i - b_i)$ $\triangleright b'_i, e'_i$ are local indexes within p

7: **if** $e'_i < 0 \vee b'_i > e_i - b_i$ **then continue**

8: $p' \leftarrow p[b'_1 : e'_1, b'_2 : e'_2, \dots, b'_N : e'_N]$ \triangleright DELEGATION to an external tool

9: $key \leftarrow (y_1, y_2, \dots, y_N)$ $\triangleright y_i \geq 0$ $\triangleright key \subset \mathbb{Z}_{\geq 0}^N$

10: $B_{new}(N) : \text{int} = \{b_1 + b'_1, b_2 + b'_2, \dots, b_N + b'_N\}$ \triangleright global indexes for p'

11: $E_{new}(N) : \text{int} = \{e_1 + e'_1, e_2 + e'_2, \dots, e_N + e'_N\}$ \triangleright within $A.d_i$

12: $C \leftarrow C \cup \{(p', key, B_{new}, E_{new})\}$ $\triangleright C$ is the set of cuts from all $p \in P$

13: $\mathbb{K} \leftarrow \mathbb{K} \cup \{key\}$ $\triangleright \mathbb{K}$ is the set of all generated merge keys

Many geospatial softwares tile 2-d arrays. However, most vendors do not publish their techniques in the research community (e.g. SciDB chunking algorithm). While not necessarily completely new to implementation, the algorithm is nevertheless new to publication: it performs tiling of arbitrarily

shaped N -d arrays, is formalized, fits ChronosServer data model, and amenable to parallelization.

4 Raster Operations

Although many raster operations exist, SciDB has very limited functionality. This section focuses on operations that are most relevant to imagery processing and supported by SciDB.

4.1 Convolution

The convolution operator $\Xi : A, K \mapsto A_{conv}$ for a 2-d array $A(d_1, d_2) : \mathbb{T}$ and a kernel $K\langle k_1, k_2 \rangle : \mathbb{T}$, where $k_i \leq |d_i|$, and $k_i \bmod 2 = 1$ for all i , produces the 2-d array $A_{conv}(d'_1, d'_2) : \mathbb{T}$ such that $d'_i = d_i[k_i \bmod 2 : |d_i| - k_i \bmod 2 - 1]$, and

$$A_{conv}[x_1, x_2] = \sum_{\substack{\forall x'_1 \in [0, k_1) \\ \forall x'_2 \in [0, k_2)}} K[x'_1, x'_2] \times A'[x'_1, x'_2] \quad (5)$$

where $A' = A[x_1 - k_1/2 : x_1 + k_1/2, x_2 - k_2/2 : x_2 + k_2/2]$, and $k_i/2 \leq x_i < |A.d_i| - k_i/2$ for all i (the division “ $/$ ” is integer).

Convolution is frequently used for satellite imagery processing [19]. For example, edge detection with the Sobel Kernels $K_1\langle 3, 3 \rangle = \{\{-1, -2, -1\}, \{0, 0, 0\}, \{1, 2, 1\}\}$ and $K_2\langle 3, 3 \rangle = \{\{-1, 0, 1\}, \{-2, 0, 2\}, \{-1, 0, 1\}\}$ happens as follows. First, local gradients are calculated at each array cell along axes d_1 and d_2 : $A_{d1} = \Xi(A, K_1)$ and $A_{d2} = \Xi(A, K_2)$, Fig. 2. Then, array $A_{edges} = \sqrt{A_{d1}^2 + A_{d2}^2}$ will contain higher values for cells classified as edges and lower values for other types of cells.

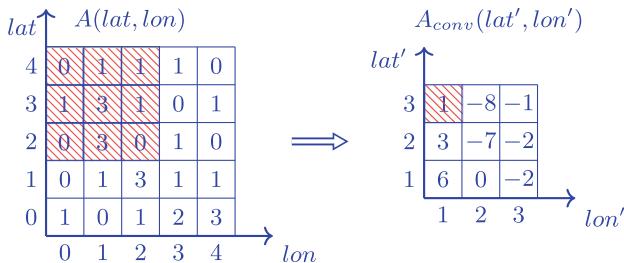


Fig. 2. An illustration of the array convolution with K_2 kernel

SciDB 16.9 (the latest version) does not have a convolution operator [11]. The closest SciDB operator to convolution is “moving window” producing the array where each cell is the result of an aggregate function (*min*, *max*, etc.)

calculated over the cells contained in an N -d rectangle around the respective cell in a source array. User-defined functions are not accepted [24].

Implementing the Sobel filter within moving window has only 3 options which are quite labor-intensive (two of them require extending SciDB by coding in C++) [24]. As a workaround, we implemented the Sobel filter using other SciDB functions. Our script contains 277 lines of code (many parts of it are similar to each other). We still think this is the easiest way to have Sobel filter in SciDB.

ChronosServer performs convolution of each 2-d $p \in P$ using **ImageMagic** assuming that algorithm 1 has been already applied to the dataset with $\rho_i \geq 1$ for $\forall i$. Since the subarrays overlap, the convolution can be applied to each subarray in parallel without data exchange between cluster nodes.

4.2 Multiresolution Pyramid

Digital maps like Google or Bing Maps display satellite imagery depending on the current map scale. First, several zoom levels are defined, e.g. $Z = \{0, 1, \dots, 16\}$. A digital map switches its current zoom level when a user zooms in/out the map. Usually, at zoom level $z \in Z$ the image resolution is $2^z \times$ less than the original one. A multiresolution pyramid is the stack of images for all zoom levels, Fig. 3. Display of downsampled images for coarser map scales significantly reduces network traffic and system load. Hence downsampling functionality is very important for an array DBMS.

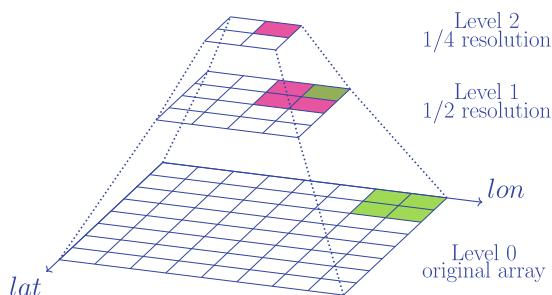


Fig. 3. Multiresolution pyramid (3 levels)

Numerous techniques exist for image downsampling [19]. However, we chose simple averaging of cells for evaluation in Sect. 5 since SciDB does not support anything more advanced. In contrast, large number of command line utilities with plethora of options exist specifically targeted at creating coarser image versions. ChronosServer delegates downsampling of a $p \in P$ to **gdalwarp**.

Formally, given a 2-d array $A(d_1, d_2) : \mathbb{T}$, its coarser version is the array $A'(d'_1, d'_2) : \mathbb{T}$ such that $A'[x'_1, x'_2] = \sum_{\forall x_i : x_i \text{ div } 2 = x'_i} A[x_1, x_2]/4$, $d'_i \langle l_i / 2 \rangle = \{(d_i[0] + d_i[1])/2, \dots, (d_i[l_i - 2] + d_i[l_i - 1])/2\}$, where $l_i = |d_i|$, Fig. 3.

At least two versions of array downsampling are possible: exact and inexact. Exact version creates A' with cell values calculated strictly according to the formulas above. In order to ensure no data exchange between the subarrays during the pyramid construction, they must be retiled beforehand such that $s_i = 2^{|Z|-1}$ for $\forall i$. Inexact version creates A' by downsampling each $p \in P$ in parallel without data exchanges. It is useful for creating quick outlooks.

4.3 Interpolation

Given an array $A(d_1, \dots, d_N) : \mathbb{T}$, a value at a coordinate (y_1, \dots, y_N) can be estimated by the operation called *interpolation*; $y_i \in (d_i[j], d_i[j+1]) \subset \mathbb{T}_i$, and $j \in [0, |d_i| - 1] \subset \mathbb{N}$. Interpolation is a core raster processing operation with many applications [19]. For example, image resolution can be increased twice by interpolation when $y_i = (d_i[j] + d_i[j+1])/2$.

As in case with downsampling, numerous interpolation techniques exist [19]. The most basic technique is the nearest neighbor: unknown cell value at a given coordinate is obtained by copying the value from the nearest cell with a known value. SciDB `xgrid` operator mimics nearest neighbor interpolation. The operator increases the length of input array dimensions by an integer scale with replication of the original values, Fig. 4. This is almost equivalent to the nearest neighbor approach since a generic interpolation must be able to increase the image resolution not only by an integer scale. ChronosServer delegates interpolation of each $p \in P$ to `gdalwarp` (no retiling is required for the nearest neighbor).

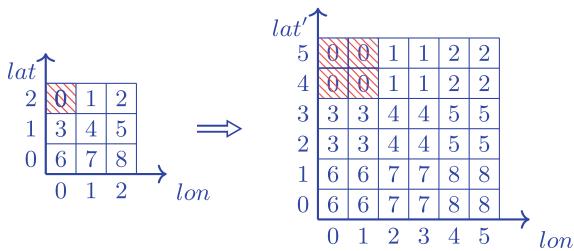


Fig. 4. Nearest neighbor interpolation (the resolution is increased twice)

5 Performance Evaluation

Microsoft Azure Cloud was taken for the experiments. Azure cluster creation, scaling up and down with given network parameters, number of virtual machines, etc. was fully automated using Java Azure SDK [22]. The latest version of Ubuntu Linux on which SciDB 16.9 runs is 14.04 LTS. We rented standard D2 v2 machines with 2 CPU cores (Intel Xeon E5-2673 v3 (Haswell) 2.4 GHz), 7 GB RAM, 100 GB local SSD drive (4 virtual data disks), max 4×500 IOPS.

A mosaic of 4×4 Landsat 8 scenes was created (band 4, paths 195–198, rows 24–27, 02–11 July 2015, 2279.94 MB in total). The average cloud cover for the scenes is 23.46%. All scenes were projected into UTM zone 31N. The resulting SciDB array shape is 24937×22855 , chunk shape is 512×512 . We used the latest SciDB version 16.9 released in November 2016. A Java program was written to convert GeoTIFF files to CSV files to feed the latter to SciDB. To date, this is the only way to import an external file into SciDB 16.9. SciDB import time of the mosaic takes ≈ 2 h on a powerful local machine. ChronosServer retiles 16 raw Landsat 8 scenes in ≈ 10 min on a single node.

Table 1 presents the results. Hyperslabbing is an extraction of a hyperslab from an array, the algorithm is in [22]. All queries persist their result as a new array object inside a DBMS. For the given operations, ChronosServer and SciDB distribute the resulting subarrays and chunks respectively between cluster nodes almost in the same way. Currently ChronosServer leaves an output subarray on the same cluster node as an input subarray. SciDB 16.9 also supports only one way of chunk distribution: `store` operator shuffles output chunks between cluster nodes using a hash function. The assignment of a chunk to a SciDB instance (a cluster node usually runs several SciDB instances) happens via a hash over chunk position minus array origin, divided by chunk size [23]. However, `xgrid` used to benchmark the interpolation inflates chunks in the way such that the output chunks are colocated with the input chunks [23]. The same assumption can be made about `regrid` operator used for creating the pyramid.

The mosaic was imported into SciDB on a local machine, the resulting SciDB array was exported into a file of proprietary SciDB binary format and copied in the Cloud (SciDB imports data from its proprietary format much faster). The area of 4×4 Landsat 8 scenes is not big data. However, very long SciDB import time prevents testing SciDB on a larger data portion.

SciDB is mostly written on C++, parameters used: 0 redundancy, 2 instances per machine, 5 execution and prefetch threads, 1 prefetch queue size, 1 operator threads, 1024 MB array cache, etc. ChronosServer has 100% Java code, ran one worker per node, OracleJDK 1.8.0_111 64 bit, max heap size 978 MB. The tools available from the standard Ubuntu 14.04 repository were used: GDAL v1.10.1 (released 2013/08/26), ImageMagick 6.7.7-10 (2017-03-14 Q16). Ubuntu 14.04 is the latest Linux version on which SciDB 16.9 is able to run.

For ChronosServer, a node contained $|P'|/M$ subarrays after “data cooking”, where M is the number of cluster nodes. SciDB also distributes data uniformly. Cold query runs were evaluated (a query is executed for the first time on given data). Every runtime reported is the average of 3 runtimes of the same query. Respective OS commands were issued to free `pagecache`, `dentries` and `inodes` each time before executing a cold query to prevent data caching at various OS levels. ChronosServer benefits from native OS caching and is much faster during hot runs when the same query is executed the second time on the same data. There is no significant runtime difference between cold and hot SciDB runs.

Table 1. Results of performance evaluation, seconds

Pyramid (3 levels)	8 nodes	16 nodes
ChronosServer	13.41	8.28
SciDB	148.57	76.28
Ratio, SciDB/Chronos	11.08	9.21
Interpolation 2×	8 nodes	16 nodes
ChronosServer	24.00	11.65
SciDB	190.83	108.36
Ratio, SciDB/Chronos	7.95	9.30
Hyperslabbing^a	8 nodes	16 nodes
ChronosServer	3.24	1.52
SciDB	5.67	3.47
Ratio, SciDB/Chronos	1.75	2.28
Sobel Filter	8 nodes	16 nodes
ChronosServer	179.22	92.71
SciDB	82087.2 ^b	7527.06 ^c
Ratio, SciDB/Chronos	458.02	81.19

^aExtract 1/4th of the image from its center

^b1/256 size of the original image

^c1/64 size of the original image; SciDB fails on full 4 × 4 scenes mosaic

6 Related Work

Numerous techniques exist for remote sensing imagery processing. This work is novel because it is in the context of array DBMS research field. Four modern raster data management trends are relevant to this paper: industrial raster data models, formal array models and algebras, in situ data processing algorithms, and array DBMS. A good survey on the algorithms is in [4].

A recent survey on array models and algebras as well as industry standard data models is in [22]. Work [22] gives the peculiar features and merits of ChronosServer data model. It is shown that the most popular array models and algebras can be mapped to Array Algebra [3]. Industry data models are also mappable to each other [14]. SciDB does not have a formal description of its data model. SciDB neither allows array dimensions to be of temporal or spatial types making it difficult or sometimes impossible to process many real-world datasets.

ChronosServer was compared only to SciDB [6] since it is the only freely available distributed array DBMS to date. The latest SciDB version does not operate in-situ and imports raster data only converted to CSV format – a very time-consuming, error-prone, and complex undertaking.

PostGIS/PostgreSQL [17] and RasDaMan [2] work on a single machine and allow registering out-database raster data in a file system. Enterprise RasDaMan version claims to be in-situ enabled and distributed but it is not freely

available [18]. To the best of our knowledge, no performance comparison between SciDB and RasDaMan has been ever published. Intel released open source TileDB on 04/04/2016. It is yet not distributed neither in-situ enabled [16, 26]. SciQL was an effort to extend MonetDB with functionality for processing multi-dimensional arrays [27]. SciQL does not provide in-situ raster processing. However, it has not yet finished nor its active development is seen so far. Commercial Oracle Spatial does not provide in-situ raster processing [15]. Commercial ArcGIS ImageServer claims in-situ raster processing with custom implementation of raster operations [1]. However, in a clustered deployment scenario all cluster nodes are recommended to hold copies of the same data or fetch data from a centralized storage. All commercial systems are not freely available.

Hadoop [9] and experimental SciDB streaming [25] allow to launch a command line tool, feed text or binary data into its standard input and ingest its standard output. Note two time-consuming data conversion phases in this case: data import into an internal database format and their conversion to other representation to be able to feed to an external software. ChronosServer directly submits files to external executables without additional data conversion steps.

There are also other efforts for in-situ and distributed raster data processing: SAGA, SWAMP, SciHadoop, SciMate, Galileo, and others. However, none of them is an array DBMS. Most of them were not released or no longer actively maintained. These systems are reviewed in [21].

7 Conclusions

ChronosServer delegates major raster data processing work to feature-rich and highly optimized command line tools. This makes it run much faster than SciDB and provide much wider functionality. At the same time, the formal array model of ChronosServer maintains a high level of independence from the underlying raster file formats and the tools.

ChronosServer is 5× to 80× faster on Landsat 8 data in GeoTIFF format than SciDB on its native storage (the same data imported into SciDB). ChronosServer data preprocessing stage takes about 10 min versus 2 h of SciDB import time not taking into account writing import code. Future work includes parallelizing the proposed tiling algorithm.

Acknowledgments. This work was partially supported by Russian Foundation for Basic Research (grant №16-37-00416).

Contributions. Rodriges: all text, figures, design and implementation of algorithms and ChronosServer, ChronosServer data model, Azure management code, SciDB import code, experimental setup. Pozdeev: SciDB cluster deployment. Bryukhov: adapted SciDB import code to Landsat data. All authors: experiments.

References

1. ArcGIS for server — Image Extension. <http://www.esri.com/software/arcgis/arcgisserver/extensions/image-extension>
2. Baumann, P., Dumitru, A.M., Merticariu, V.: The array database that is not a database: file based array query answering in rasdaman. In: Nascimento, M.A., Sellis, T., Cheng, R., Sander, J., Zheng, Y., Kriegel, H.-P., Renz, M., Sengstock, C. (eds.) SSTD 2013. LNCS, vol. 8098, pp. 478–483. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40235-7_32
3. Baumann, P., Holsten, S.: A comparative analysis of array models for databases. *Int. J. Database Theory Appl.* **5**(1), 89–120 (2012)
4. Blanas, S., Wu, K., Byna, S., Dong, B., Shoshani, A.: Parallel data analysis directly on scientific file formats. In: ACM SIGMOD 2014, pp. 385–396 (2014)
5. Coverity scan: GDAL. <https://scan.coverity.com/projects/gdal>
6. Cudre-Mauroux, P., et al.: A demonstration of SciDB: a science-oriented DBMS. *Proc. VLDB Endowment* **2**(2), 1534–1537 (2009)
7. Earth on AWS. <https://aws.amazon.com/earth/>
8. GeoTIFF. <http://trac.osgeo.org/geotiff/>
9. Hadoop streaming. <wiki.apache.org/hadoop/HadoopStreaming>
10. ImageMagic: History. <http://imagemagick.org/script/history.php>
11. Interpolation - SciDB forum. <http://forum.paradigm4.com/t/interpolation/1283>
12. Landsat apps. <https://aws.amazon.com/blogs/aws/start-using-landsat-on-aws/>
13. Landsat project statistics. <https://landsat.usgs.gov/landsat-project-statistics>
14. Nativi, S., Caron, J., Domenico, B., Bigagli, L.: Unidatas common data model mapping to the ISO 19123 data model. *Earth Sci. Inf.* **1**, 59–78 (2008)
15. Oracle spatial and graph. <http://www.oracle.com/technetwork/database/options/spatialandgraph/overview/index.html>
16. Papadopoulos, S., et al.: The TileDB array data storage manager. *Proc. VLDB Endowment* **10**, 349–360 (2016)
17. PostGIS raster data management. http://postgis.net/docs/manual-2.2/using-raster_dataman.html
18. RasDaMan features. <http://www.rasdaman.org/wiki/Features>
19. Richards, J.A.: Remote Sensing Digital Image Analysis: An Introduction, 5th edn. Springer, Heidelberg (2013). <https://doi.org/10.1007/978-3-642-30062-2>
20. Rodriges Zalipynis, R.A.: Chronoserver: real-time access to “native” multi-terabyte retrospective data warehouse by thousands of concurrent clients. *Inf. Cybern. Comput. Eng.* **14**(188), 151–161 (2011)
21. Rodriges Zalipynis, R.A.: ChronosServer: fast in situ processing of large multidimensional arrays with command line tools. In: Voevodin, V., Sobolev, S. (eds.) RuSCDays 2016. CCIS, vol. 687, pp. 27–40. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-55669-7_3
22. Rodriges Zalipynis, R.A.: Distributed in situ processing of big raster data in the Cloud. In: Perspectives of System Informatics - 11th International Andrei Ershov Informatics Conference, PSI 2017, Moscow, Russia, June 27–29, 2017, Revised Selected Papers. LNCS. Springer (2017, in press)
23. SciDB output chunk distribution. <http://forum.paradigm4.com/t/does-store-redistributes-chunks-among-cluster-nodes/1919>

24. SciDB window functions. <http://forum.paradigm4.com/t/user-defined-window-functions/1790>
25. SciDB streaming. <https://github.com/Paradigm4/streaming>
26. TileDB. <http://istc-bigdata.org/tiledb/index.html>
27. Zhang, Y., et al.: SciQL: bridging the gap between science and relational DBMS. In: IDEAS (2011)

Impulsive Noise Removal from Color Images with Morphological Filtering

Alexey Ruchay^(✉) and Vitaly Kober

Department of Mathematics, Chelyabinsk State University, Chelyabinsk, Russia
ran@csu.ru

Abstract. This paper deals with impulse noise removal from color images. The proposed noise removal algorithm employs a novel approach with morphological filtering for color image denoising; that is, detection of corrupted pixels and removal of the detected noise by means of morphological filtering. With the help of computer simulation we show that the proposed algorithm can effectively remove impulse noise. The performance of the proposed algorithm is compared in terms of image restoration metrics and processing speed with that of common successful algorithms.

Keywords: Color image · Impulsive noise removal · Denoising
Morphological filtering

1 Introduction

Color image processing has received much attention in the last years [1]. Digital image processing algorithms are generally sensitive to noise. A color image is treated as a mapping $Z_2 \rightarrow Z_3$ that assigns to a point $x = (i, j)$ on the image plane a three-dimensional vector (x^r, x^g, x^b) , where the superscripts correspond to the red, green, and blue color image channels. In this way, a color image is considered as a two-dimensional vector field, and each vector has three color components.

The most popular algorithms for removal of impulsive noise in color images utilize the ordering of pixels belonging to a local window W [2], and assign a dissimilarity measure to each color pixel from the window. Several switching techniques are proposed [3, 4] to adapt parameters of filters to the processed image. A switching algorithm verifies the following hypothesis: is the central pixel of window W affected by noise? If the central pixel is corrupted by noise then it is replaced by the output of a local robust filter; otherwise, it is left unchanged (see Fig. 1). One of efficient switching schemes is referred to as the sigma vector median filter (SVMF) [4].

The performance of a switching filtering depends mainly on the impulse noise detection. If the detector fails to identify corrupted pixels, the performance of the algorithm yields errors of missed impulse noise. On the other hand, if the detector

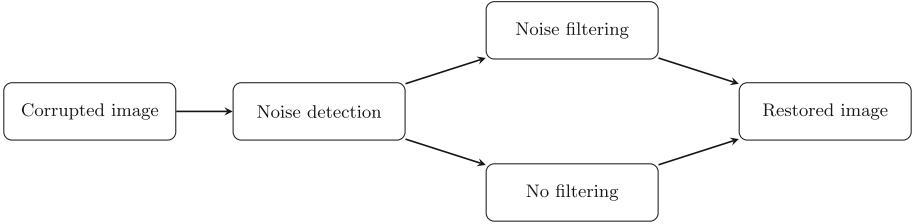


Fig. 1. Switching filtering scheme.

wrongly identifies uncorrupted pixels as noisy, the performance of the algorithm yields false impulse noise errors. In the both cases the overall performance of image restoration is poor. The performance of switching filtering algorithms can be compared with various image restoration measures [3, 4]. In this paper, type I and type II errors are used to characterize the performance of tested algorithms. A type I error occurs when the algorithm asserts something that is absent, a false hit. A type I error is called false positive (FP). A type II error occurs when the algorithm fails to assert what is present, a miss. A type II error is called false negative (FN).

Mathematical morphology describes the shape and structure of certain objects, and is used to extract the useful components in the image. It is utilized for image filtering, image segmentation, image measurement, area filling and so on [5, 6]. In the image denoising aspect, we can get fairly good effect by applying the gray morphology, having the characteristics of nonlinearity and parallelism [7, 8].

In this paper a novel approach to color image denoising by morphological filtering is proposed. With the help of computer simulation we show that the proposed algorithm can effectively remove impulse noise. The performance of the proposed algorithm is compared in terms of image restoration metrics with that of common successful algorithms [9].

The paper is organized as follows. In Sect. 2, we describe the proposed switching algorithm by morphological filtering. Section 3 describes impulsive noise models. Computer simulation results are provided in Sect. 4. Finally, Sect. 5 summarizes our conclusions.

2 Proposed Algorithm

A common impulse noise removal algorithm is based on the reduced vector ordering, which assigns a dissimilarity measure to each color pixel x_i from the local window $W = \{x_1, x_2, \dots, x_n\}$ of the size $n = 9$. Let $\rho(x_i, x_j)$ be the distance between two vectors x_i, x_j , then the inner product is defined as

$$d_i = \sum_{j=1}^n \rho(x_i, x_j), x_j \in W. \quad (1)$$

The meaning of the product is the distance associated with the central pixel x_i inside the filtering window W . The ordering of the distances as $d_{(1)} \leq d_{(2)} \leq \dots \leq d_{(n)}$, implies the same ordering to the corresponding vectors x_j $x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(n)}$.

The original value of the central pixel x_1 in the window W is being replaced by $x_{(1)}$ which means that

$$x_{(1)} = \operatorname{argmin}_{x_i \in W} d_i. \quad (2)$$

This concept of replacing is a common way to define the mean scale in vector spaces. It is called the Vector Median Filter (VMF) [10]. Most commonly the L_2 metric is used for the design of the VMF

$$\rho(x_i, x_j) = \left(\sum_{k=1}^3 (x_i^k - x_j^k)^2 \right)^{1/2}.$$

2.1 Rank Weighted Vector Median Filter

The reduced ordering schemes are based on the sum of the dissimilarity measures between a given pixel and all other pixels from the filtering window W [3]. In this way, the output of the VMF is the pixel whose average distance to other pixels is minimized.

The distances $d_{ij} = \|x_i - x_j\|$ between the pixel x_i and all other pixels x_j belonging to W , ($j = 1, \dots, n$) can be ordered as $d_{i1}, d_{i2}, \dots, d_{in} \rightarrow d_{i(1)} \leq d_{i(2)} \leq \dots \leq d_{i(n)}$, and the ranks of the ordered distances can be used for building the cumulative distances in Eq. (1).

Let r denote the rank of a given distance, and $d_{i(r)}$ stand for the corresponding distance value. So, instead of the aggregated distances in Eq. (1) we can build a weighted sum of distances, utilizing the distance ranks as

$$\Delta_i = \sum_{i=1}^n f(r) d_{i(r)},$$

where $f(r)$ is a decreasing weighting function of the distance rank r , like $f(r) = 1$, $f(r) = 1/r$ and $f(r) = 1/r^2$. The $f(r) = 1/r$ weights for the design of the adaptive switching filter is recommended in [3].

Then, the rank weighted sum of distances calculated for each pixel belonging to W can be sorted and a new sequence of vectors can be obtained $\Delta_1, \Delta_2, \dots, \Delta_n \rightarrow x_{(1)}^* \leq x_{(2)}^* \leq \dots \leq x_{(n)}^*$, where the vector $x_{(1)}^*$ is the output of the rank weighted vector median filter (RWVMF) [3].

Similarly to Eq. (2) the RWVMF output $x_{(1)}^*$ can be defined as

$$x_{(1)}^* = \operatorname{argmin}_{x_i \in W} \sum_{r=1}^n f(r) d_{i(r)}.$$

The structure of the switching filter is defined [3] as follows. If the difference $\Delta_1 - \Delta_{(1)}$ exceeds a threshold value α , then a pixel is declared as corrupted by a impulsive noise; otherwise, it is treated as uncorrupted

$$y_1 = \begin{cases} x_{AMF}, & \text{if } \Delta_1 - \Delta_{(1)} > \alpha, \\ x_1, & \text{otherwise,} \end{cases} \quad (3)$$

where y_1 is the switching filter output, x_1 is the central pixel of the filtering window W and x_{AMF} is the Arithmetic Mean Filter (AMF) output computed over the pixels declared by the detector as uncorrupted. Extensive experiments revealed that very good denoising results can be achieved using the following switching filter:

$$y_1 = \begin{cases} x_{VMF}, & \text{if } \Delta_1 - \Delta_{(1)} > \alpha, \\ x_1, & \text{otherwise,} \end{cases}$$

where x_{VMF} is the standard VMF output computed for all the pixels in the filtering window W .

Detection noise method `DetectionMethod1` for pixel x_1 of the filtering window W in RWVMF in Eq. (3) can be defined as follows:

$$\text{DetectionMethod1}(x_1) = \begin{cases} 1, & \text{if } \Delta_1 - \Delta_{(1)} > \alpha, \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

where 1 means the successful detection of noise and 0 means no noise detected.

We propose the following modification of the noise detection given in Eq. (4). This detection noise method `DetectionMethod2` for pixel x_1 of the filtering window W in RWVMF can be defined as follows:

$$\text{DetectionMethod2}(x_1) = \begin{cases} 1, & \text{if } \Delta_{(1)} > \alpha, \\ 0, & \text{otherwise.} \end{cases}$$

The detection method `DetectionMethod1` and `DetectionMethod2` use the predefined parameter α .

2.2 Fast Peer Group Filter

Recently, a peer group filter has been proposed [11, 12]. The peer group associated with the central pixel of the window denotes a set of such pixels whose distance to the central pixel does not exceed a predefined threshold. The Fast Peer Group Filter (FPGF) replaces the center of the filtering window with the VMF output when a specified number of the smallest distances between the central pixel and its neighbors differ not more than a predefined threshold.

Let vector components $x_i \in [0, 1]$ represent the color channel values in a given color space quantified into the integer domain. In the first step, the size of the peer group, or in other words, the number of close neighbors of the central pixel of the filtering window x_1 is determined. A pixel $x_i \neq x_1$ belonging to W is

a close neighbor of x_1 , if the normalized Euclidean distance $d(x_i, x_1)$ in a given color space is less than a predefined threshold valued $d \in [0, 1]$.

In the RGB color space, the peer group size denoted as m_k is the number of pixels from W contained in a sphere with radius d centered at pixel x_k . $m_k = \#\{x_j \in W : \|x_j - x_k\| < d\}$, where $\#$ denotes the cardinality and $\|\cdot\|$ stands for the Euclidean norm.

If the peer group size of the central pixel x_1 of the filtering window W is $m_1 \leq 2$, then this pixel is treated as an outlier. The structure of the switching filter can be defined as follows:

$$y_1 = \begin{cases} x_{VMF}, & \text{if } m_k \leq k, \\ x_1, & \text{otherwise,} \end{cases} \quad (5)$$

where x_{VMF} is standard VMF output computed for all the pixels of window W , and k is a parameter that determines the minimal size of the peer group.

Detection noise method `DetectionMethod3` for the pixel x_1 of the window W in Eq. (5) can be defined as follows:

$$\text{DetectionMethod3}(x_1) = \begin{cases} 1, & \text{if } m_k \leq k, \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

The detection method `DetectionMethod3` has parameter d and k .

We propose a modification of the detection noise method `DetectionMethod3` in Eq. (6). The proposed method `DetectionMethod4` utilizes iteratively the detection noise method `DetectionMethod3`. At first step `DetectionMethod3` is used with parameters $d = 0.25$ and $k = 3$. This step corresponds to a preliminary detection of noise. Then, the `DetectionMethod3` is iteratively used with modified parameters d and k . Experiments showed that good denoising results can be achieved using the proposed detection method.

2.3 Morphological Filter

Morphological processing is constructed with operations on sets of pixels. Binary morphology uses only set membership and is indifferent to the value, such as gray level or color, of a pixel. We will deal here only with morphological operations for binary images. Therefore we use a threshold operation `BW(A, level)` to convert the grayscale image A to a binary image. The output image replaces all pixels in the input image with than the threshold `level` by 1 (white) and replaces all other pixels with 0 (black).

The operation intersection $A \cap B$ produces a set that contains the elements in both A and B . The operation union $A \cup B$ produces a set that contains the elements of both A and B . The complement A^c is the set of elements that are not contained in A . The difference of two sets A and B , denoted by $A - B$ is $A \cap B^c$. A standard morphological operation is the reflection of all of the points in a set \hat{A} about the origin of the set A .

Dilation and erosion are basic morphological processing operations. Let A be a set of pixels and let B be a structuring element. Let $(\widehat{B})_s$ be the reflection of B about its origin and followed by a shift by s . Dilation operation is the set of all shifts that satisfy the following: $A \oplus B = \{s | ((\widehat{B})_s \cap A) \subseteq A\}$. Erosion operation is the set of all shifts that satisfy the following: $A \ominus B = \{s | (B)_s \subseteq A\}$.

Closing operation is a dilation followed by an erosion: $A \circ B = (A \oplus B) \ominus B$. Opening operation is an erosion followed by a dilation: $A \bullet B = (A \ominus B) \oplus B$.

Morphological “bottom hat” operation is an image minus the morphological closing of an image: $A \star B = ((A \oplus B) \ominus B) - A$.

Morphological “remove” operation is a removing interior pixels of an image A , written as `remove(A)`. This operation sets a pixel to 0 if all its 4-connected neighbors are 1, thus leaving only the boundary pixels on.

Let a color image X be three-dimensional vector (x^r, x^g, x^b) each channel is processed individually. We propose the following `DetectionMethod5(X)` method of the noise detection for color image X with a morphological filter. The output of this method is

$$M = M_1 \cup M_2 \cup M_3 \cup M_4 \cup M_5.$$

$$M_1 = \bigcup_{i=r,g,b} (\text{set1}(x^i, mset) \star B) \bigcup_{i=r,g,b} (\text{set2}(x^i, pset) \star B),$$

$$M_2 = \text{remove} \left(\bigcup_{i=r,g,b} \text{BW}(\text{set2}(x^i, pset), \text{level}) \right),$$

$$M_3 = \text{remove} \left(\bigcup_{i=r,g,b} \text{BW}(\text{set3}(x^i, mset), \text{level}) \right),$$

$$M_4 = (\text{BW}(\text{rgb2gray}(X), \text{level}) \star B),$$

$$M_5 = (\text{BW}(\text{rgb2gray}(\text{set2}(X, pset)), \text{level}) \star B),$$

where B is the standard structuring element, `set1(A, mset)` is subtraction of all the pixels A value `mset`, `set2(A, pset)` is subtraction of the values `pset` of all the pixels A , `set3(A, mset)` is addition of all the pixels A value `mset`, `rgb2gray` is conversion of the color image X to the grayscale intensity image.

The detection method `DetectionMethod5` uses the parameters: `pset`, `mset`, `level`.

3 Model of Impulse Noise

Color images may be contaminated by various types of impulse noise [3, 10, 13, 14]. Impulse noise corruption often occurs in digital image acquisition or transmission process as a result of photo-electronic sensor faults or channel bit errors. Image transmission noise may be caused by various sources, such as car ignition systems, industrial machines in the vicinity of the receiver, switching

transients in power lines, lightning in the atmosphere and various unprotected switches. This type of transmission noise is often modeled as impulse noise. Let us consider models of impulse noise used for computer simulation. Let X_i be the vector characterizing a pixel of a noisy image, q be the vector describing one of the noise models, x_i be the noise-free color vector, p be the probability of impulse noise occurrence. Each tested image can be corrupted with different probabilities, that is, $p \in \{0.1, 0.2, 0.3\}$. Depending on the type of vector q , either fixed-valued or random-valued impulse noise models are considered.

Assume that channels are corrupted independently (CI). So, we use the following models of impulse noise:

$$X_i = \begin{cases} (q_1, x_i^g, x_i^b), & \text{with probability } p(1-p)^2, \\ (x_i^r, q_2, x_i^b), & \text{with probability } p(1-p)^2, \\ (x_i^r, x_i^g, q_3), & \text{with probability } p(1-p)^2, \\ (q_1, q_2, x_i^b), & \text{with probability } p^2(1-p), \\ (x_i^r, q_2, q_3), & \text{with probability } p^2(1-p), \\ (q_1, x_i^g, q_3), & \text{with probability } p^2(1-p), \\ (q_1, q_2, q_3), & \text{with probability } p^3, \\ (x_i^r, x_i^g, x_i^b), & \text{with probability } (1-p)^3, \end{cases}$$

where q_1, q_2, q_3 are spatially uniform distributed independent random variables with the probability of p . The corrupted pixels can be defined in different manner; that is, CI1 means that they take values of either 0 or 255; CI2 means that corrupted pixel is a random variable with uniform distribution in the interval of $[0, 255]$; CI3 means that corrupted pixel is a random variable with uniform distribution in the intervals of $[0, 55]$ and $[200, 255]$. Additionally, we introduce a model CT when all channels of the color image are contaminated simultaneously by impulsive noise as follows:

$$X_i = \begin{cases} (q_1, q_2, q_3), & \text{with probability } p, \\ (x_i^r, x_i^g, x_i^b), & \text{with probability } (1-p). \end{cases}$$

The corrupted pixels can be defined in different manner as CT1, CT2, CT3.

4 Computer Simulation

The performance of the detection methods `DetectionMethod(1-5)` is compared with respect to FP and FN errors. Since FP and FN errors depend on the parameters of the detection methods, then the receiver operating characteristic (ROC) curve as a function of FP and FN errors is utilized. The parameters of detection methods can be chosen from the ROC curve to provide the minimum FP and FN errors. Minimum FP and FN errors for all tested methods `DetectionMethod(1-5)` with the type of noise CI1-3, CT1-3, $p = 0.1, 0.2, 0.3$ are summarized in Table 1. One can observe that the proposed method

DetectionMethod5 detects noise very well comparing with other detection techniques. The algorithm of the removal of impulsive noise by a switching filter can be defined as follows:

$$y_1 = \begin{cases} x_{VMF}, & \text{if } \text{DetectionMethod}(x_1) = 1, \\ x_1, & \text{otherwise,} \end{cases}$$

where x_{VMF} is the standard VMF output computed for all the pixels of the window W .

Table 1. Minimum FP and FN errors for DetectionMethod(DM) 1–5 with type of noise (TN) CI1-3, CT1-3, $p = 0.1, 0.2, 0.3$.

TN	FPDM1	FNDM1	FPDM2	FNDM2	FPDM3	FNDM3	FPDM4	FNDM4	FPDM5	FNDM5
CI1 0.1	0.037	0.094	0.094	0.062	0.067	0.099	0.021	0.075	0.033	0
CI2 0.1	0.100	0.145	0.099	0.107	0.115	0.212	0.063	0.123	0.113	0.263
CI3 0.1	0.090	0.107	0.097	0.087	0.121	0.155	0.048	0.092	0.075	0.034
CT1 0.1	0.016	0.013	0.085	0.017	0.043	0.007	0.004	0.009	0.03	0
CT2 0.1	0.042	0.022	0.086	0.036	0.089	0.062	0.027	0.043	0.111	0.001
CT3 0.1	0.049	0.043	0.086	0.042	0.078	0.064	0.027	0.044	0.062	0.004
CI1 0.2	0.143	0.098	0.104	0.109	0.312	0.065	0.05	0.088	0.037	0
CI2 0.2	0.175	0.134	0.118	0.144	0.181	0.207	0.083	0.112	0.205	0.218
CI3 0.2	0.173	0.101	0.108	0.125	0.184	0.136	0.064	0.102	0.092	0.028
CT1 0.2	0.022	0.086	0.060	0.083	0.152	0.044	0.007	0.071	0.131	0
CT2 0.2	0.023	0.047	0.062	0.096	0.054	0.052	0.010	0.034	0.158	0.026
CT3 0.2	0.031	0.110	0.062	0.099	0.078	0.110	0.023	0.083	0.068	0.004
CI1 0.3	0.397	0.087	0.159	0.132	0.659	0.036	0.167	0.107	0.043	0
CI2 0.3	0.289	0.237	0.15	0.304	0.280	0.311	0.161	0.261	0.14	0.446
CI3 0.3	0.294	0.153	0.067	0.240	0.438	0.079	0.163	0.122	0.158	0.024
CT1 0.3	0.041	0.112	0.098	0.084	0.342	0.009	0.020	0.065	0.028	0
CT2 0.3	0.041	0.113	0.103	0.102	0.169	0.030	0.024	0.075	0.082	0.023
CT3 0.3	0.048	0.210	0.106	0.111	0.197	0.138	0.033	0.172	0.087	0.005

We use the mean square error (MSE) and the peak signal to noise ratio (PSNR) as measures of restoration quality. They are defined as

$$MSE = \frac{1}{3N} \sum_{i=1}^N \sum_{k=1}^3 (x_i^k - y_i^k)^2, \quad PSNR = 20 \log_{10} \left(\frac{255}{\sqrt{MSE}} \right),$$

where x_i^k , $k = 1, 2, 3$ are the component of the original image, and y_i^k are the restored components.

In order to provide comparison of noise removal techniques taking into account subjective human evaluation, we use FSIMc [15], SR-SIM [16] and IFS [17] quality metrics which are suitable for inspection of color images. The results of impulsive noise removal presented in Tables 2 and 3 show that proposed method **DetectionMethod5** with morphological filtering achieves the best performance with respect to the all considered quality color image metrics.

Table 2. Comparison of efficiency of denoising by switching filter based on Detection-Method(DM) 1–5 using quality measure (QM) PSNR, MSE, IFS, FSIM, SRSIM with type of noise CI1-3, CT1-3, $p = 0.1, 0.2$.

QM	CI1 0.1	CI2 0.1	CI3 0.1	CT1 0.1	CT2 0.1	CT3 0.1	CI1 0.2	CI2 0.2	CI3 0.2
PSNR DM1	28.986	27.685	27.832	30.692	30.259	29.145	24.262	24.732	24.470
PSNR DM2	28.529	30.154	29.715	30.345	31.808	30.947	23.318	25.531	24.721
PSNR DM3	28.294	27.770	27.843	30.411	28.881	28.822	23.395	24.559	24.734
PSNR DM4	29.852	30.570	29.983	32.408	31.853	31.600	25.687	25.620	25.689
PSNR DM5	32.025	27.754	30.835	34.254	31.052	31.664	27.482	24.784	27.042
MSE DM1	82.109	110.78	107.10	55.447	61.252	79.159	243.70	218.69	232.31
MSE DM2	91.234	62.748	69.433	60.050	42.878	52.275	302.83	181.94	219.24
MSE DM3	96.306	108.65	106.85	59.141	84.117	85.278	297.55	227.59	218.58
MSE DM4	67.270	60.363	66.304	37.343	42.723	51.624	175.53	178.23	175.44
MSE DM5	40.798	109.067	53.659	24.418	51.041	44.328	116.128	216.119	128.499
IFS DM1	0.9636	0.9493	0.9532	0.9796	0.9717	0.9578	0.9005	0.9025	0.9029
IFS DM2	0.9576	0.9675	0.9668	0.9780	0.9754	0.9643	0.8932	0.9194	0.9155
IFS DM3	0.9546	0.9498	0.9509	0.9754	0.9568	0.9501	0.8918	0.9040	0.9113
IFS DM4	0.9690	0.9687	0.9685	0.9860	0.9887	0.9677	0.9256	0.9199	0.9228
IFS DM5	0.9756	0.9514	0.9696	0.9876	0.9656	0.9699	0.9437	0.9141	0.9337
FSIM DM1	0.9789	0.9669	0.9680	0.9865	0.9834	0.9730	0.9378	0.9349	0.9380
FSIM DM2	0.9772	0.9767	0.9749	0.9856	0.9860	0.9775	0.9330	0.9422	0.9435
FSIM DM3	0.9744	0.9669	0.9654	0.9856	0.9753	0.9681	0.9316	0.9302	0.9424
FSIM DM4	0.9823	0.9769	0.9753	0.9912	0.9864	0.9783	0.9542	0.9436	0.9520
FSIM DM5	0.9845	0.9694	0.9769	0.9928	0.977	0.9785	0.9622	0.9407	0.9564
SRSIM DM1	0.9903	0.9827	0.9835	0.9946	0.9911	0.9869	0.9754	0.9713	0.9734
SRSIM DM2	0.9905	0.9895	0.9895	0.9944	0.9932	0.9900	0.9737	0.9762	0.9770
SRSIM DM3	0.9884	0.9829	0.9832	0.9933	0.9872	0.9838	0.9695	0.9681	0.9748
SRSIM DM4	0.9921	0.9897	0.9896	0.9967	0.9935	0.9921	0.9812	0.9770	0.9795
SRSIM DM5	0.9932	0.9869	0.9921	0.9978	0.9893	0.9984	0.9842	0.9749	0.9808

The result of denoising based on the proposed detection method **DetectionMethod5** presented in Figs. 2, and 3. We see that the proposed method with morphological filtering yields good results in terms of objective and subjective criteria.

Next we provide execution time of denoising algorithms with switching filter based on **DetectionMethod5** with type of noise CI1-3, CT1-3, $p = 0.1, 0.2, 0.3$. 20 experiments were carried out and the results are averaged. Table 4 show that the proposed algorithm with morphological filtering yields the best results in terms of execution time.

Table 3. Comparison of efficiency of denoising by switching filter based on Detection-Method(DM) 1–5 using quality measure (QM) PSNR, MSE, IFS, FSIM, SRSIM with type of noise CI1-3, CT1-3, $p = 0.2, 0.3$.

QM	CT1 0.2	CT2 0.2	CT3 0.2	CI1 0.3	CI2 0.3	CI3 0.3	CT1 0.3	CT2 0.3	CT3 0.3
PSNR DM1	23.572	25.547	25.126	18.246	19.246	20.556	18.393	20.720	19.674
PSNR DM2	23.026	24.637	25.683	19.206	19.245	20.316	19.426	21.052	22.297
PSNR DM3	26.004	25.753	26.528	16.929	18.738	20.153	21.077	22.704	23.182
PSNR DM4	26.436	26.573	26.893	20.368	19.533	21.808	21.404	22.717	23.574
PSNR DM5	27.626	25.242	28.549	22.666	18.522	23.262	23.715	22.362	25.01
MSE DM1	285.64	181.28	199.74	973.69	773.49	572.01	941.25	550.90	700.85
MSE DM2	323.95	223.51	175.67	780.66	773.58	604.55	741.98	510.38	383.16
MSE DM3	163.16	172.87	144.62	1318.5	869.44	627.63	507.42	348.84	312.50
MSE DM4	155.95	143.13	132.95	597.33	723.92	428.82	502.49	329.36	352.54
MSE DM5	112.32	194.49	90.82	351.93	913.89	306.84	276.43	377.44	205.15
IFS DM1	0.8963	0.8961	0.8931	0.7523	0.6922	0.7904	0.7687	0.8296	0.7655
IFS DM2	0.8887	0.8820	0.8981	0.7833	0.6799	0.8189	0.8099	0.8441	0.8371
IFS DM3	0.9358	0.9037	0.9122	0.7598	0.6741	0.7975	0.8519	0.8771	0.8577
IFS DM4	0.9367	0.9163	0.9267	0.8186	0.6972	0.8500	0.8539	0.8783	0.8592
IFS DM5	0.9393	0.8968	0.9341	0.8824	0.6946	0.884	0.9029	0.8739	0.8964
FSIM DM1	0.9289	0.9349	0.9307	0.8373	0.8015	0.8663	0.8317	0.9084	0.8320
FSIM DM2	0.9277	0.9235	0.9398	0.8510	0.8006	0.8640	0.8599	0.9076	0.8929
FSIM DM3	0.9564	0.9436	0.9414	0.8106	0.7824	0.8562	0.8862	0.9247	0.9077
FSIM DM4	0.9566	0.9482	0.9456	0.8778	0.8211	0.8882	0.8867	0.9254	0.9095
FSIM DM5	0.9584	0.9273	0.9496	0.9154	0.8194	0.9052	0.9265	0.9136	0.9187
SRSIM DM1	0.9752	0.9769	0.9701	0.9312	0.8973	0.9448	0.9315	0.9512	0.9289
SRSIM DM2	0.9751	0.9724	0.9735	0.9377	0.8957	0.9431	0.9446	0.9472	0.9570
SRSIM DM3	0.9817	0.9788	0.9725	0.9136	0.8752	0.9369	0.9557	0.9584	0.9623
SRSIM DM4	0.9830	0.9826	0.9777	0.9517	0.9125	0.9561	0.9563	0.9621	0.9625
SRSIM DM5	0.9877	0.9708	0.9782	0.9651	0.9122	0.9582	0.9728	0.9512	0.9688

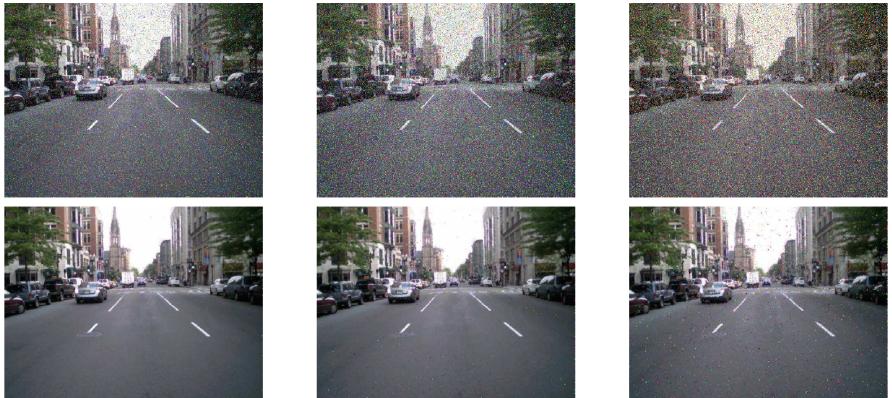


Fig. 2. Results of denoising by switching filter based on DetectionMethod5 with type of noise CI1 $p = 0.1, 0.2, 0.3$.



Fig. 3. Results of denoising by switching filter based on DetectionMethod5 with type of noise CT1 $p = 0.1, 0.2, 0.3$.

Table 4. Execution time (seconds) of denoising algorithms based on DetectionMethod(DM) 1-5 with type of noise CI1-3, CT1-3, $p = 0.1, 0.2, 0.3$.

DM	DM1	DM2	DM3	DM4	DM5
Execution time	16.12	16.02	7.43	7.23	0.06

5 Conclusion

In the paper, new noise detection techniques for switching filtering of impulse noise with morphological filtering were proposed. Computer simulation performed on test images contaminated by six noise models revealed a very high efficiency of the proposed method. The performance of the proposed algorithm was evaluated in terms of objective and subjective criteria of image restoration. With the help of computer simulation we showed that the proposed algorithm with morphological filtering can effectively remove impulse noise. Moreover the proposed algorithm is the faster among all tested algorithms.

Acknowledgements. The work was supported by the Ministry of Education and Science of Russian Federation, grant 2.1743.2017.

References

1. Kober, V.: Robust and efficient algorithm of image enhancement. *IEEE Trans. Consum. Electron.* **52**(2), 655–659 (2006)
2. Dinet, E., Robert-Inacio, F.: Color median filtering: a spatially adaptive filter. In: *Proceedings of Image and Vision Computing New Zealand*, pp. 71–76 (2007)
3. Smolka, B., Malik, K., Malik, D.: Adaptive rank weighted switching filter for impulsive noise removal in color images. *J. Real-Time Image Proc.* **10**, 289–311 (2015)

4. Lukac, R., Smolka, B., Plataniotis, K., Venetsanopoulos, A.: Vector sigma filters for noise detection and removal in color images. *J. Vis. Commun. Image Represent.* **17**(1), 1–26 (2006)
5. Soille, P.: *Morphological Image Analysis: Principles and Applications*, 2nd edn. Springer, New York (2003). <https://doi.org/10.1007/978-3-662-05088-0>
6. Najman, L., Talbot, H.: *Mathematical Morphology: From Theory to Applications*. ISTE-Wiley, London (2010)
7. Jakhar, A., Sharma, S.: A novel approach for image enhancement using morphological operators. *IJARCST* **2**, 300–302 (2014)
8. Yoshitaka, K.: Mathematical morphology-based approach to the enhancement of morphological features in medical images. *J. Clin. Bioinf.* **1**, 33 (2011)
9. Ruchay, A., Kober, V.: Clustered impulse noise removal from color images with spatially connected rank filtering, vol. 9971, pp. 99712Y–99712Y-10 (2016)
10. Khryashchev, V., Kuykin, D., Studenova, A.: Vector median filter with directional detector for color image denoising. In: *Proceedings of the World Congress on Engineering*, vol. 2, pp. 1–6 (2011)
11. Smolka, B., Chydzinski, A.: Fast detection and impulsive noise removal in color images. *J. Real Time Imaging* **11**(5–6), 389–402 (2005)
12. Malinski, L., Smolka, B.: Fast averaging peer group filter for the impulsive noise removal in color images. *J. Real-Time Image Proc.* **11**, 427–444 (2016)
13. Singh, K., Bora, P.: Adaptive vector median filter for removal of impulse noise from color images. *J. Electr. Electron. Eng.* **4**(1), 1063–1072 (2004)
14. Venkatesan, P., Nagarajan, G.: Removal of Gaussian and impulse noise in the colour image progression with fuzzy filters. *Int. J. Electron. Sig. Syst.* **3**(1), 1–6 (2013)
15. Zhang, L., Zhang, L., Mou, X., Zhang, D.: FSIM: a feature similarity index for image quality assessment. *IEEE Trans. Image Process.* **20**(8), 2378–2386 (2011)
16. Zhang, L., Li, H.: SR-SIM: a fast and high performance IQA index based on spectral residual. In: *19th IEEE International Conference on Image Processing (ICIP)*, pp. 1473–1476 (2012)
17. Chang, H., Zhang, Q., Wu, Q., Gan, Y.: Perceptual image quality assessment by independent feature detector. *Neurocomputing* **151**, 1142–1152 (2015)

Optimization Problems on Graphs and Network Structures

An Exact Polynomial Algorithm for the Outerplanar Facility Location Problem with Improved Time Complexity

Edward Gimadi^{1,2(✉)}

¹ Sobolev Institute of Mathematics, 4 Koptyuga av., 630090 Novosibirsk, Russia
gimadi@math.nsc.ru

² Novosibirsk State University, 2 Pirogova Str., 630090 Novosibirsk, Russia

Abstract. The Unbounded Facility Location Problem on outerplanar graphs is considered. The algorithm with time complexity $O(nm^3)$ was known for solving this problem, where n is the number of vertices, m is the number of possible plant locations. Using some properties of maximal outerplanar graphs (binary 2-trees) and the existence of an optimal solution with a family of centrally-connected service areas, the recurrence relations are obtained allowing to design an algorithm which can solve the problem in $O(nm^{2.5})$ time.

Keywords: Outerplanar graph · Exact algorithm · Time complexity · Dynamic programming

1 Introduction

One of the important mathematical models of the clustering problem is the Unbounded Facility Location Problem (UFLP) on graphs and networks [4, 10, 11, 13]. In this model, it is required to find a suitable partition of a certain set of elements of the lower level (consumers) into clusters served by top-level elements (suppliers).

The UFLP can be formulated as follows [11]: minimize the function of total costs

$$\sum_{i \in M} f_i x_i + \sum_{j \in V} \sum_{i \in M} b_j c_{ij} x_{ij} \quad (1)$$

subject to

$$\sum_{i \in M} x_{ij} = 1, \quad j \in V, \quad (2)$$

$$x_{ij} \leq x_i, \quad i \in M, \quad j \in V, \quad (3)$$

$$x_{ij}, x_i \in \{0, 1\}, \quad (4)$$

where

M is the set of possible locations of suppliers, $|M| = m$;

V is the set of consumers, $|V| = n$;

b_j is the size of demand at the site j ;

f_i is the fixed cost of establishing of the supplier on the site i ;

c_{ij} is the transportation cost of the product unit from the supplier on the site i to the consumer j ;

x_i and x_{ij} are variables of a choice and an assignment respectively.

A more compact formulation of the UFLP can be written as follows: find the minimum of the function

$$\sum_{i \in S} f_i + \sum_{j \in V} b_j \min_{i \in S} c_{ij} \quad (5)$$

over all nonempty subsets S of M .

For the UFLP analysis it is another convenient compact formulation which uses the supplier *assignment vector* π as a variable: minimize the function

$$\sum_{i \in I(\pi)} f_i + \sum_{j \in V} b_j c_{\pi_j j} \quad (6)$$

over all vectors $\pi = (\pi_1, \dots, \pi_n)$, where $\pi_j \in M$ is the index of the supplier-site serving the customer-site $j \in V$ and $I(\pi)$ is the set of plants included in a solution π .

In the general case, the problem that arises is NP-hard in the strong sense, since the Vertex Cover Problem [5] reduces to it evidently. This stimulates the search for polynomially solvable particular cases of the problem. In the article such a case is presented in the form of the network UFLP. Nevertheless the UFLP on an arbitrary network to be NP-hard also [11].

The *network UFLP* is defined by means of a simple connected undirected weighted graph $G = (V, E)$ with the vertex set V of the sites (consumers) and the edge set of communications connecting these sites. It is assumed that $M \subset V$ and that the transportation cost c_{ij} is equal to the sum of edge weights (lengths) in a shortest path connecting sites i and j (the distance between i and j). It is known that The N

The *tree-network UFLP* was solved in $O(n^3)$ -time by Trubin [12]. Later the same algorithm was rediscovered by Kolen [9].

In [6] it was established that

Statement 1. *The tree-network UFLP can be solved in $O(nm)$ time.*

This time does not exceed $O(n^2)$ certainly. Later similar $O(n^2)$ time algorithms for solving the tree-network UFLP were presented in [3,11].

The algorithm of [6] uses the notion of *connected service areas*.

A service area $A \in V$ will be called *connected with respect to the graph* $G = (V, E)$ if the subgraph induced by A is connected.

Denote by

$$i \leq_v k, \ i <_v k, \ i =_v k$$

the relations $g_{iv} \leq g_{kv}$, $g_{iv} < g_{kv}$, and $g_{iv} = g_{kv}$, respectively. Notations

$$i \leq_{V'} k, \quad i <_{V'} k, \quad \text{and } i =_{V'} k$$

mean that respective relations hold for every $v \in V'$, $V' \subset V$. We say that a matrix $(g_{ij})(i \in M, j \in V)$ possesses *a connectedness property with respect to an acyclic network G* if for each pair $i, k \in M$ there exists a partition (V', V'') such that the subgraphs induced by V' and V'' are connected and the following relations hold:

$$i \leq_{V'} k, \quad i <_{V''} k.$$

In [2] the connectedness property of the matrix (g_{ij}) was reformulated as the notion of a matrix $(g_{ij})(i \in M, j \in V)$ *connected with respect to an arbitrary graph G*.

The next claim uses the notion of *central connectedness* [7] of the transportation matrix. A matrix (g_{ij}) is said to be *centrally-connected with respect to a network G (briefly, C-matrix)*, if $g_{i_1, v} < g_{i_2, v}$ for all $i_1, i_2 \in M, v \in V$ implies $g_{i_1, j} < g_{i_2, j}$ for all sites j on a shortest path connecting the sites i_1 and v .

Statement 2 [7]. *For an arbitrary network UFLP with C-matrix (g_{ij}) , there exists an optimal solution with a family of centrally-connected service areas.*

In this case the network UFLP is solved in $O(nm^2 + |E|)$ time, if the network contains only pseudo-tree quasiblocks, and in $O(n^2m)$ time, if the number of possible suppliers locations in each not pseudo-tree quasiblock does not exceed $\log n$ [7].

An example of C-matrix is a matrix with components $g_{ij} = c_i + \tilde{c}_{ij}$, where c_i are arbitrary vertex weights and \tilde{c}_{ij} is the distance between vertices i and j .

In this paper we consider a class of location problems on outerplanar graphs.

By definition, *an outerplanar graph* is a graph that has a planar drawing for which all vertices belong to the outer face of the drawing. The outerplanar graphs are subgraphs of parallel-series graphs. The maximum outerplanar graphs are graphs, to which one can not add an edge without loss of outerplanarity. This is exactly 2-trees [14].

Ageev [1] designed a polynomial-time transformation from the UFLP with the matrix connected with respect to outerplanar graphs to the UFLP with the matrix connected with respect to cycles and, as a consequence, designed an algorithm for this problem with running time $O(n^3m)$. In [2] the UFLP on partial 2-trees (including series-parallel networks) was solved in $O(nm^3)$ -time using the technique similar to the algorithm for the tree-network UFLP [6, 7].

Somewhat earlier using different techniques Hassin and Tamir presented an $O(nm^3)$ -time algorithm for the UFLP on series-parallel networks [8].

So in the case of known algorithms with the linear (relative to the cardinality n of the vertex set) running time there is an essential gap between $O(nm)$ time and $O(nm^3)$ time of the algorithms for the UFLP on trees and on 2-trees respectively.

Below we present an algorithm solving the UFLP on outerplanar graphs, in which the expression m^3 in evaluation of time complexity is replaced by the expression $m^{2.5}$.

2 Main Result and Preliminary Considerations

The main statement of the presented paper is the following result:

Theorem 1. *An optimal solution of UFLP on outerplanar networks can be found in $O(nm^{2.5})$ time.*

In order to prove this statement, we will present several recurrence relations allowing to design an algorithm which solves the outerplanar UFLP in $O(nm^{2.5})$ time.

Note that the outerplanar UFLP can be reduced to the UFLP on a maximal outerplanar graph, adding at most $(n - 3)$ new edges with large enough lengths. A maximal outerplanar graph on n vertices has $(2n - 3)$ edges.

We will call an edge of a maximal outerplanar graph *outer* if it is adjacent to exactly one triangle and *inner* otherwise.

Below it will be convenient to use an alternative definition of a maximal outerplanar graph, namely, a *binary 2-tree*. An undirected graph G without cut vertices we will call a *2-tree* if either G is a triangle, or it can be constructed from any its triangle by means consecutive connection to one of its edge (p, q) two new edges $(p, s), (s, q)$ with a new vertex s . A *binary 2-tree* is a 2-tree whose every edge is adjacent to at most two triangles.

Choose as the *root edge* of the given graph an outer edge $e_1 \in E$.

Let $V_{pq} \subset V$ denote the set of vertices-descendants of the edge (p, q) (except the endpoints of this edge). For each $v \in V_{pq}$, the edge (p, q) is contained in a minimal sequence of edge-coupled triangles joining the vertex v and the root edge $e_1 \in E$.

Set $N_{pq} = |V_{pq}|; V_{pq}^0 = V_{pq} \cup \{p\} \cup \{q\}; M_{pq} = V_{pq} \cap M$.

Assign the numbers $\{1, \dots, n\}$ to nodes of G in the counterclockwise order on the outer face so that the chosen edge e_1 is labeled as $(1, n)$. Moreover $p < q$ for all edges (p, q) .

Note that the set V_{pq}^0 coincides with the segment $[p, q]$.

For every inner edge (p, q) (and outer edge e_1), denote by $Son(p, q)$ the only vertex $s \in [p, q]$ such that there are edges (p, s) and (s, q) . For $s = Son(p, q)$, we will denote by $L(s)$ and $R(s)$ nodes p and q , respectively.

Consider the family of the following SPLP on subgraphs of the original binary 2-tree:

$$\left\{ G_{pq}; i, j \mid \pi_p = i, \pi_q = j \right\}, \quad 1 \leq p < q \leq n; \quad i, j \in M, \quad (7)$$

if we do not take into account the costs f_i, f_j, g_{ip}, g_{jq} . Here G_{pq} is the subgraph induced by the vertex set defined by the segment $[p, q]$. Note that G_{pq} may be not the binary 2-tree, but if $(p, q) \in E$, G_{pq} is a binary 2-tree.

Let $\{F_{pq}(i, j)\}$ be optima of corresponding problems (7). Then using the notation

$$f_k^{ij} = \begin{cases} 0 & \text{if } k \in \{i, j\}, \\ f_k & \text{otherwise,} \end{cases}$$

for every $i, j, k \in M$, the optimum of the original problem is equal to

$$F^* = \min_{i \in M} \left\{ f_i + g_{i1} + \min_{j \in M} \{ f_j^{ii} + g_{jn} + F_{1,n}(i, j) \} \right\}.$$

Lemma 1. *For any vertex $s \in V$ (with $p = L(s), q = R(s)$) and a pair $(i, j) \in M$, the following recurrence relations hold:*

$$F_{pq}(i, j) = \min \left\{ D_{pq}(i, j), \min_{k \in \{i, j\}} \{ F_{ps}(i, k) + g_{ks} + F_{sq}(k, j) \} \right\},$$

where

$$D_{pq}(i, j) = \min_{k \in M_{pq}} \left\{ F_{ps}(i, k) + (f_k + g_{ks}) + F_{sq}(k, j) \right\}. \quad (8)$$

Proof. Correctness of the statement follows from the existence of a central-connected optimal solution by Statement 2 and a representation of the graph G_{pq} with $(p, q) \in E$ in a form of two subgraphs G_{ps} and G_{sq} , connected by the edge (p, q) and the vertex $s = Son(p, q)$.

Lemma 1 is proved.

Using these relations we can solve the outerplanar FLP within the same time bound $O(nm^3)$ as in [2, 8]. The time complexity of the algorithm depends mainly on the calculation of values $D_{pq}(i, j)$. Below we present a more efficient way for computing the values $D_{pq}(i, j)$.

3 On Some Property of Binary 2-Trees

We continue with auxiliary property of binary trees. Namely, given an integer r , $0 \leq r < n/2$ we establish an upper bound for the cardinality of the set

$$V(n, r) = \left\{ s \in V \mid N_{L(s)s} \geq r, N_{sR(s)} \geq r \right\}.$$

Lemma 2. *The following inequalities hold:*

$$|V(n, r)| \leq \frac{n-1}{r+1} - 1, \quad 0 \leq r < n/2. \quad (9)$$

Proof. It is clear that the statement is true for the minimal 2-tree with $n = 3$. Let the conclusion of the statement hold for the binary 2-trees with less than n vertices. In a binary n -vertex 2-tree $G = G(n)$ select the binary 2-trees $G(n_1) = G_{1s}$ and $G(n_2) = G_{sn}$, induced by the vertex sets V_{1s}^0 and V_{sn}^0 respectively, where s means $Son(1, n)$. Setting $n_1 = |V_{1s}^0|$, $n_2 = |V_{sn}^0|$ we have the relation $n_1 + n_2 - 1 = n$. This relation and the inequalities (9) for the graphs $G(n_1)$ and $G(n_2)$ imply that

$$\begin{aligned} |V(n, r)| &\leq |V(n_1, r)| + |V(n_2, r)| + 1 \\ &\leq \left(\frac{n_1-1}{r+1} - 1 \right) + \left(\frac{n_2-1}{r+1} - 1 \right) + 1 = \frac{n_1+n_2-2}{r+1} - 1 = \frac{n-1}{r+1} - 1. \end{aligned}$$

Lemma 2 is proved.

4 Computing $D_{pq}(i, j)$

4.1 CASE $Son(p, q) \in V(n, r)$

Lemma 3. *A collection of values*

$$\left\{ D_{pq}(i, j) \mid Son(p, q) \in V(n, r), (p, q) \in E, i, j \in M \right\} \quad (10)$$

can be calculated in $O(nm^3/r)$ time.

Proof. For fixed $i, j \in M$ and $(p, q) \in E$, the value $D_{pq}(i, j)$ determined by (8) is calculated in $O(m)$ time. By Lemma 2 the number of edges (p, q) with $Son(p, q) \in V(n, r)$ does not exceed n/r . Taking into account all pairs i, j of suppliers we obtain the required time bound.

Lemma 3 is proved.

4.2 CASE $Son(p, q) \notin V(n, r)$

Lemma 4. *A collection of values*

$$\left\{ D_{pq}(i, j) \mid Son(p, q) \notin V(n, r), (p, q) \in E, i, j \in M \right\}$$

can be found in $O(nm^2r + nmr^3)$ time.

Proof. Fix $s \in V(n, r)$ and put $p = L(s)$, $q = R(s)$. By the statement of the Lemma either $N_{ps} < r$ or $N_{sq} < r$. say that first inequality holds.

For any $i, j \in M$ represent the expression (8) in the following form:

$$D_{pq}(i, j) = \min \{ D_{pq}^L(i, j), D_{pq}^R(i, j) \},$$

where

$$D_{pq}^L(i, j) = \min_{p < k < s} \left\{ F_{ps}(i, k) + (f_k + g_{ks}) + F_{sq}(k, j) \right\}. \quad (11)$$

$$D_{pq}^R(i, j) = \min_{s \leq k < q} \left\{ F_{ps}(i, k) + (f_k + g_{ks}) + F_{sq}(k, j) \right\}. \quad (12)$$

By Lemma 1 the collection of values

$$\left\{ D_{pq}^L(i, j) \mid (p, q) \in E, i, j \in M \right\}$$

can be calculated in $O(nm^2r)$ time.

Now we need a proper way for finding the collection of values

$$\left\{ D_{pq}^R(i, j) \mid (p, q) \in E, i, j \in M \right\} \quad (13)$$

in case $N_{ps} < r$ (which means $s \leq p + r$).

In order to complete the proof of Lemma 4 we have to show the following

Lemma 5. *The collection (13) can be calculated in $O(nmr^3)$ time.*

Proof. The proof of Lemma 5 is based on the following two facts.

Fact 1. *For every edge $(p, q) \in E$ and a pair $i, j \in M$, the following recurrence relations hold:*

$$D_{pq}^R(i, j) = \min_{p < v < s} \{ \mathcal{F}'_{vs}(i) + \mathcal{F}''_{vs}(j) \}, \quad (14)$$

where $s = Son(p, q)$ and

$$\mathcal{F}'_{vs}(i) = \min_{p \leq k' < s} \left\{ F_{pv}(i, k') + f_{k'}^{ii} + g_{k'v} \right\}, \quad (15)$$

$$\mathcal{F}''_{vs}(j) = \min_{s \leq k < q} \left\{ g_{k,v+1} + F_{v+1,s}(k, k) + f_k + g_{ks} + F_{sq}(k, j) \right\}. \quad (16)$$

Proof. For a vertex $s \in \{p, q\}$, denote by $F_{pq}^{(s)}(i, j)$ the optimum of an objective for the UFLP on the outerplanar graph induced by the vertex set $\{p, p+1, \dots, q\}$ provided that $\pi_p = i, \pi_q = j$ and without costs f_{π_s} and g_{π_ss} .

Then the value $F_{ps}(i, k)$ in (12) can be represented as follows:

$$F_{ps}(i, k) = \min_{p < v < s} \left\{ \min_{p < k' \leq v} F_{pv}^{(p)}(i, k') + F_{v+1,s}^{(s)}(k, k) \right\}.$$

Hence (12) can be written as

$$D_{pq}^R(i, j) = \min_{s \leq k < q} \left\{ \min_{p < v < s} \left\{ \min_{p < k' \leq v} F_{pv}^{(p)}(i, k') + F_{v+1,s}^{(s)}(k, k) \right\} + (f_k + g_{ks}) + F_{sq}(k, j) \right\}.$$

Changing the order of minima over k and v we obtain the following expression:

$$D_{pq}^R(i, j) = \min_{p < v < s} \left\{ \min_{p < k' \leq v} F_{pv}^{(p)}(i, k') + \min_{s \leq k < q} \{ F_{v+1,s}^{(s)}(k, k) + (f_k + g_{ks}) + F_{sq}(k, j) \} \right\}.$$

Finally since

$$F_{pv}^{(p)}(i, k') = F_{pv}(i, k') + f_{k'}^{ii} + g_{k'v},$$

$$F_{v+1,s}^{(s)}(k, k) = g_{k,v+1} + F_{v+1,s}(k, k),$$

the expression $D_{pq}^R(i, j)$ can be written in the form (14)–(16). The proof of Fact 1 is complete.

Note that (15) does not depend on the supplier j and (16) does not depend on the supplier i . Each of them can take m values. Instead of those relations we obtain a dependence on v and on k' respectively. The number of different values of v (and of k') does not exceed r .

Fact 2. Both collections of values

$$\left\{ \mathcal{F}'_{vs}(i) \mid L(s) < v < s, s \notin V(n, r), i \in M \right\} \quad (17)$$

and

$$\left\{ \mathcal{F}''_{vs}(j) \mid L(s) < v < s, s \notin V(n, r), j \in M \right\} \quad (18)$$

can be calculated in $O(nmr^3)$ time.

Proof. Consider the collections (17) and (18) separately.

Put $p = L(s)$. Let $]a, b[$ be the segment without the endpoints a and b .

Finding the collection (17) requires $O(mnc^2)$ time if we already know the values $F_{vs}(i, k')$ for all $v, k' \in]p, s[$. A part of these values was found already: namely, for $v, k' \in]p, s'[$, where $s' = Son(p, s)$. The remaining values $F_{pv}(i, k')$ for $v, k' \in]s', s[$ we can calculate (using the values $F_{s'v}(i', k')$ for $v, i', k' \in]s', s[$ which we already have) by means of the following recurrence relation:

$$F_{pv}(i, k') = \min_{p < i' < v} \left\{ F_{ps'}(i, i') + f_{i'}^{ik'} + g_{i's'} + F_{s'v}(i', k') \right\}.$$

This can be done in total $O(nmr^3)$ time.

Calculation of (18) has complexity if we already have the necessary values $F_{v+1,s}(k, k)$ for all $v \in]p, s[$ and $k \in]s, q[$. They can be found in $O(nmr)$ time with the help of the following recurrence relation:

$$F_{v,s}(k, k) = g_{k,v} + F_{v,R(v)}(k, k) + F_{R(v),s}(k, k),$$

where $(v, R(v)) \in E$, $v \in]p, s[$, and $F_{s,s}(k, k) = 0$.

Hence all necessary values (16) can be found in $O(nmr^2)$ time.

Both collections (17) and (18) require $O(nmr^3)$ time for their calculation, completing the proof of Fact 2.

So Lemma 5 and furthermore Lemma 4 are proved.

5 Proof of the Main Result

It follows from Lemmas 3 and 4 that we can find the values $D_{pq}(i, j)$, for every $(p, q) \in E$, $i, j \in M$, in $O(nm\psi_{mr})$ time, where

$$\psi_{mr} = m^2/r + mr + r^3.$$

Setting $r = \lfloor \sqrt{m} \rfloor$ we obtain the upper bound $O(m^{1.5})$ for the value ψ_{mr} and the exact algorithm with time complexity $O(nm^{2.5})$ for solving the Outerplanar UFLP. This completes the proof of Theorem 1 and the paper.

Acknowledgments. The author was supported by the Russian Science Foundation, project no. 16-11-10041.

References

1. Ageev, A.A.: Graphs, matrices and the simple plant location problem (in Russian). *Upravlyayemye Sistemy* **29**, 3–12 (1989)
2. Ageev, A.A.: A polynomial algorithm for solving the location problem on a series-parallel network (in Russian). *Upravlyayemye Sistemy* **30**, 3–6 (1990)
3. Billonnet, A., Costa, M.-C.: Solving the uncapacitated plant location problem on trees. *Discrete Appl. Math.* **49**(1–3), 51–59 (1994)
4. Gadegaard, S.L.: Discrete Location Problems. A Ph.D. dissertation, 150 p. Aarhus University Department of Economics and Business Economics (2016)
5. Garey, M.R., Johnson, D.S.: Computers and Intractability. Freeman, San Francisco (1979). 338 p
6. Gimadi, E.K.: An efficient algorithm for solving plant location problem with service regions connected with respect to an acyclic network (in Russian). *Upravlyayemye Sistemy* **23**, 12–23 (1983)
7. Gimadi, E.K.: The problem of location on a network with centrally connected service areas (in Russian). *Upravlyayemye Sistemy* **25**, 38–47 (1984)
8. Hassin, R., Tamir, A.: Efficient algorithm for optimization and selection on series-parallel graphs. *SIAM J. Algebraic Discrete Methods* **7**(3), 379–389 (1986)
9. Kolen, A.: Solving covering problems and the uncapacitated plant location on the trees. *Eur. J. Oper. Res.* **12**(3), 266–278 (1983)
10. Laporte, G., Nickel, S., da Gama, F.S. (eds.): Location Science. Springer, Cham (2015). <https://doi.org/10.1007/978-3-319-13111-5>. 644 p.
11. Mirchandani, P.B., Francis, R.L. (eds.): Discrete Location Theory. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley, New York/Chichester/Brisbane/Toronto/Singapore (1990). 555 p
12. Trubin, V.A.: An efficient algorithm for plant locating on trees (in Russian). *Dokl. AN SSSR* **231**(3), 547–550 (1976)
13. Ulukan, Z., Demircioglu, E.: A survey of discrete facility location problems. *Int. J. Soc. Behav. Educ. Econ. Bus. Ind. Eng.* **9**(7), 2487–2492 (2015)
14. Valdes, J., Tarjan, R.E., Lawler, E.L.: The recognition of series parallel digraphs. *SIAM J. Comput.* **11**(2), 298–313 (1982)

Approximation Algorithms for the Maximum m -Peripatetic Salesman Problem

Edward Kh. Gimadi^{1,2} and Oxana Yu. Tsidulko^{1,2(✉)}

¹ Sobolev Institute of Mathematics, 4 Acad. Koptyug Avenue,
630090 Novosibirsk, Russia

gimadi@math.nsc.ru, tsidulko.ox@gmail.com

² Novosibirsk State University, 2 Pirogova Str., 630090 Novosibirsk, Russia

Abstract. We consider the maximum m -Peripatetic Salesman Problem (MAX m -PSP), which is a natural generalization of the classic Traveling Salesman Problem. The problem is strongly NP-hard. In this paper we propose two polynomial approximation algorithms for the MAX m -PSP with different and identical weight functions, correspondingly. We prove that for random inputs uniformly distributed on the interval $[a, b]$ these algorithms are asymptotically optimal for $m = o(n)$. This means that with high probability their relative errors tend to zero as the number n of the vertices of the graph tends to infinity. The results remain true for the distributions of inputs that minorize the uniform distribution.

Keywords: Maximum m -Peripatetic Salesman Problem
Time complexity · Edge-disjoint Hamiltonian cycles

1 Introduction

We consider the maximum m -Peripatetic Salesman Problem (MAX m -PSP), where given a complete undirected n -vertex graph $G = (V, E)$ and m weight functions $w_i : E \rightarrow \mathbb{R}_+$, $i = 1, \dots, m$, one for each salesman, the problem is to find m edge-disjoint Hamiltonian cycles $H_1, \dots, H_m \subset E$ that maximize their total weight:

$$\sum_{i=1}^m w_i(H_i) = \sum_{i=1}^m \sum_{e \in H_i} w_i(e).$$

Recall that Hamiltonian cycle is a simple cycle that contains all vertices of the graph. If all m weight functions are identical, we will refer to the problem as the MAX m -PSP with *identical weight functions*, or simply MAX m -PSP as is common in the literature. If all the weight functions differ from each other, the problem is referred to as the MAX m -PSP with *different weight functions*.

The m -PSP was first introduced by Krarup in 1975 [19]. It was shown that the problem of finding two edge-disjoint Hamiltonian cycles is NP-complete [6]. Hence, it follows that the 2-PSP is NP-hard both in the maximization and minimization variants. These results can be extended to the general m -PSP, $m > 2$.

The most common application of the m -PSP is optimizing delivery routes or design of transportation loops for automated guided vehicles serving manufacturing cells in a factory, where in order to avoid congestion edge-disjoint loops are preferred [8]. Another example is the construction of patrol routs, where to improve safety and efficiency several edge disjoint routes that maximize the gathered information are used [8]. The m -PSP also arise in a network design application where several edges-disjoint cycles are determined in order to protect the network from link failure [7].

The m -PSP is a natural generalization of the well-known NP-hard Traveling Salesman Problem (TSP). The goal of the TSP is to find only one Hamiltonian cycle of optimal total weight in the given graph. Note that the optimal Hamiltonian cycle corresponds to a cyclic permutation or, in other words, ordering of the vertices of the graph, so each time we need to place items in an optimal order the TSP arise. For example, the TSP was used as a step in Abstract Meaning Representation-to-text generation problem to place the fragments of the AMR-graph in an optimal order [22]. Another application is the superstring problem, which is given a set of strings find the shortest superstring that contains each given string as a substring. The problem can be reduced to the MAX TSP in the overlap graph [17]. A widespread MAX TSP application in data analysis is the matrix reordering. Given a large object-feature matrix in which the rows correspond to objects and the columns correspond to some numerical features of the objects, the problem is to reorder the rows or the columns of the matrix to maximize the sum of similarities between adjacent rows or columns. This gives a visualization of patterns in data. Similar columns that are adjacent in the reordered matrix can indicate the dependent or duplicated features. Similarity of the rows provides ideas for clustering of the objects. The latter property was studied for clustering and ordering human genes, proteins, web-users [5, 16, 21], and also for the problem of compressing and storing data [15]. The solution of the MAX m -PSP, in turn, gives m disjoint permutations or m different orderings, so for each object we can obtain a set of m most similar objects.

As it was mentioned before, the m -PSP is NP-hard, which means that in solving the problem we have to choose between fast and exact algorithms. We are interested in constructing polynomial approximation algorithms with proven performance guarantees.

The most studied is the case of 2-PSP. For the symmetric MAX 2-PSP algorithms with guaranteed approximation ratios $3/4$ and $7/9$ were designed in [1, 14]. Paper [12] propose polynomial approximation algorithms for the 2-PSP with weights in a given interval $[1, q]$ and for the MAX 2-PSP with weights 1 and 2. For the metric MAX m -PSP, where the weighs of edges satisfy the triangle inequality, a $5/6$ -approximation algorithm is given in [13]. Paper [3] presents an asymptotically optimal algorithm with running-time $O(n^3)$ for the Euclidean MAX m -PSP.

In [10, 11] two approaches to asymptotically optimal solving the MIN m -PSP on random inputs in polynomial time are developed. In this paper we discuss how these approaches can be modified for the MAX m -PSP in the case of uniformly

distributed random inputs and possess even better performance guarantees, that don't depend on the boundaries of the distribution domain.

To describe the quality of the algorithms we will use the $(\varepsilon_A(n), \delta_A(n))$ notation. An approximation algorithm A for a maximization problem has *performance guarantees* $\varepsilon_A(n)$ and $\delta_A(n)$ on the set of random inputs of the problem of size n , if

$$\Pr\left\{F_A(I) < (1 - \varepsilon_A(n))OPT(I)\right\} \leq \delta_A(n), \quad (1)$$

where $F_A(I)$ and $OPT(I)$ are the approximate and the optimum value of the objective function of the problem on the input I , respectively, $\varepsilon_A(n)$ is an assessment of the *relative error* of the solution obtained by algorithm A, $\delta_A(n)$ is an estimation of the *failure probability* of the algorithm, which equals to the proportion of cases when the algorithm fails, i.e. it does not hold the relative error $\varepsilon_A(n)$ or doesn't produce any answer at all.

An algorithm A is called *asymptotically optimal* on the class of instances of the problem, if there exist performance guarantees such that $\varepsilon_A(n) \rightarrow 0$ and $\delta_A(n) \rightarrow 0$ as $n \rightarrow \infty$.

2 Algorithm for the MAX m -PSP with Different Weight Functions

To solve the MAX m -PSP with different weight functions we will use the greedy algorithm \tilde{A}_1 from [10] modified for the maximum case of the problem. Algorithm \tilde{A}_1 consists of Stages $i = 1, \dots, m$; $m < n/4$, where at i -th Stage we build the i -th Hamiltonian cycle H_i .

Stage i of Algorithm \tilde{A}_1 :

For a fixed i consider graph $G(V, E)$ with the remaining edges and the i -th weight function of edges $w_i : E \rightarrow \mathbf{R}_+$. Set the partial path $P_i = \{u_1\}$, $s = 1$.

Step 1 (Greedy). Let $P_i = \{u_1, \dots, u_s\}$ be the constructed partial path. If $s = n - 4i$, go to Step 2. Otherwise find an edge $(u_s, u_{s+1}) \in G \setminus P_i$ with maximum weight and add it to the path $P_i = \{u_1, \dots, u_s, u_{s+1}\}$. Set $s = s + 1$ and return to the beginning of Step 1.

Step 2 (Extension-rotation). Consider an undirected graph $H = (V_H, E_H)$ with a vertex set $V_H = (V \setminus P_i) \cup \{u_s, u_1\}$, and the set of edges E_H containing all edges of G between the vertices of V_H .

Note that the minimum degree of a vertex in H is at least $|V_H|/2$, so according to Dirac's Theorem H contains a Hamiltonian path, which can be found by standard *extension-rotation* procedure. At each step the procedure grows the partial path (v_1, \dots, v_ℓ) as follows. If possible, it adds a new edge from the end-vertex v_ℓ to a vertex outside the constructed path. Otherwise, it rotates the path, i.e. it adds an edge from the end-vertex v_ℓ to a vertex v_i in the path and deletes the edge (v_i, v_{i+1}) making vertex v_{i+1} the new end of the path.

Applying the extension-rotation procedure build a Hamiltonian path $\{u_s = v_1, v_2, \dots, v_{n-s}, v_{n-s+1} = u_1\}$ in graph H .

Finally, set cycle

$$H_i = \{u_1, \dots, u_s, v_2, \dots, v_{n-s}, u_n, u_1\} = \{u_1^{(i)}, u_2^{(i)}, \dots, u_n^{(i)}, u_1^{(i)}\}.$$

Delete all edges of the constructed H_i from G , so they won't be used in building cycles $H_j, j > i$, thus all cycles will be edge-disjoint. Go to Stage $i + 1$.

As a result we have m edge-disjoint Hamiltonian cycles of total weight $F_{\tilde{A}_1} = \sum_{i=1}^m \sum_{e \in H_i} w_i(e)$. The construction of each Hamiltonian cycle takes $O(n^2)$ time, so the time complexity of Algorithm \tilde{A}_1 is $O(mn^2)$.

2.1 Probabilistic Analysis of Algorithm \tilde{A}_1

Our goal here is to show the conditions under which the algorithm \tilde{A}_1 is asymptotically optimal. Let the weights $w_{ijk} = w_i(j, k)$ of edges of the input graph be independent identically distributed (i.i.d.) random reals with uniform distribution $\text{UNI}[a, b]$ on $[a, b]$. Let the random variable $\xi_{is} = w_i(u_s^{(i)}, u_{s+1}^{(i)})$ be the weight of the s -th edge of the i -th Hamiltonian cycle constructed by algorithm \tilde{A}_1 . According to (1), the performance guarantees $(\varepsilon_n, \delta_n)$ of algorithm \tilde{A}_1 for MAX m -PSP are determined by the inequality:

$$\Pr \left\{ \sum_{i=1}^m \sum_{s=1}^{n-4i-1} \xi_{is} + \sum_{i=1}^m \sum_{s=n-4i}^n \xi_{is} < (1 - \varepsilon_{\tilde{A}_1}(n)) OPT \right\} \leq \delta_{\tilde{A}_1}(n). \quad (2)$$

Here the first sum consists of the weights of the edges that were chosen at the greedy Step 1 of \tilde{A}_1 , and the second sum corresponds to Step 2. Consider a term ξ_{is} from the first sum in (2). It is equal to maximum of at least $n - 2(i - 1) - s$ independent reals with $\text{UNI}[a, b]$ distribution, since for each vertex 2 incident edges were deleted from G at each of previous $(i - 1)$ Stages, and when adding the s -th edge to the partial path P_i at Step 1, s vertices of the graph already belong to P_i . We estimate the weights ξ_{is} from the second sum as $\xi_{is} \geq a$.

Now we normalize the initial weights of edges as $w'_{ijk} = (b - w_{ijk})/(b - a) \in [0, 1]$, and set random variables $\xi'_{is} = (b - \xi_{is})/(b - a)$. If ξ_{is} is a maximum weight of an edge chosen at the greedy step of algorithm \tilde{A}_1 for a problem with the set of weights (w_{ijk}) , then the ξ_{is} is the corresponding minimum edge weight for a problem with the set of weights (w'_{ijk}) . Using $OPT \leq bnm$ we get the upper bound for (2):

$$\begin{aligned} & \Pr \left\{ \sum_{i=1}^m \sum_{s=1}^{n-4i-1} \xi_{is} + \sum_{i=1}^m \sum_{s=n-4i}^n \xi_{is} < (1 - \varepsilon_{\tilde{A}_1}) OPT(I) \right\} \\ & \leq \Pr \left\{ \sum_{i=1}^m \sum_{j=1}^{n-4i-1} \frac{b - \xi_{is}}{b - a} + \sum_{i=1}^m \sum_{s=n-4i}^n \frac{b - a}{b - a} > \frac{bmn - (1 - \varepsilon_{\tilde{A}_1}) bmn}{b - a} \right\} \\ & \leq \Pr \left\{ \sum_{i=1}^m \sum_{s=1}^{n-4i-1} \xi'_{is} > mn\varepsilon_{\tilde{A}_1} - m(2m + 3) \right\} \leq \delta_{\tilde{A}_1}. \end{aligned} \quad (3)$$

The elements of the sum in (3) are independent random variables, thus their sum can be estimated using Petrov's Theorem [20, Chap. 2.2].

Theorem 1. [20] Let X_1, \dots, X_k be independent random variables. Let there be positive constants g_1, \dots, g_k and T , such that

$$\mathbf{E} \exp(tX_j) \leq \exp(g_j t^2/2)$$

for $j = 1, \dots, k$, $0 \leq t \leq T$. Then

$$\Pr\left\{\sum_{i=1}^k X_j > x\right\} \leq \begin{cases} \exp\left(-\frac{x^2}{2G}\right) & \text{if } 0 \leq x \leq GT, \\ \exp\left(-\frac{Tx}{2}\right) & \text{if } x \geq GT. \end{cases}$$

where $\mathbf{E}X$ is the expected value of random variable X , $G = \sum_{i=1}^k g_j$.

In [10] the estimations were made for a probabilistic inequality similar to (3).

Theorem 2. [10] Let the random variables $\xi'_{is} \in (0, 1)$ be defined as above. Then for

$$\Pr\left\{\sum_{i=1}^m \sum_{s=1}^{n-4i-1} \xi'_{is} > mne^{\varepsilon'} a/b - m(2m+3)\right\} \leq \delta'$$

using Theorem 1, we can estimate (ε', δ') as:

- (1) $\varepsilon' = O\left(\frac{b_n/a_n}{n/\ln n}\right)$, $\delta' = n^{-9}$, if $2 \leq m < \ln n$;
- (2) $\varepsilon' = O\left(\frac{b_n/a_n}{n^\theta}\right)$, $\delta' = n^{-9}$, if $\ln n \leq m \leq n^{1-\theta}$.

Using Theorem 2 with $\varepsilon_{\tilde{A}_1} = a\varepsilon'/b$ for (3) we obtain the following result.

Theorem 3. In the case of random inputs with distribution $UNI[a, b]$, $0 \leq a \leq b$, algorithm \tilde{A}_1 for the MAX m -PSP gives asymptotically optimal solutions with performance guarantees, that don't depend on the boundaries of $[a, b]$:

- (1) $\varepsilon_{\tilde{A}_1} = O\left(\ln n/n\right)$, $\delta_{\tilde{A}_1} = n^{-9}$, if $2 \leq m < \ln n$;
- (2) $\varepsilon_{\tilde{A}_1} = O\left(n^{-\theta}\right)$, $\delta_{\tilde{A}_1} = n^{-9}$, if $\ln n \leq m \leq n^{1-\theta}$.

3 Algorithm for the MAX m -PSP with Identical Weight Functions

The probability analysis of the algorithm from the previous section crucially depends on the use of Petrov's Theorem and the independence of different weight functions. This is clearly not the case for the MAX m -PSP with *identical* weight functions. Therefore we cannot expect the algorithm \tilde{A}_1 to give good results for this variant of the problem.

In this section we propose algorithm \tilde{A}_2 for MAX m -PSP with identical or different weight functions. Algorithm \tilde{A}_2 consists of the following three steps.

Step 1. Uniformly split the initial complete n -vertex graph G into subgraphs G_1, \dots, G_m , so that $V(G_i) = V(G)$, and for each edge e in $E(G)$ choose with equal probability $1/m$ a subgraph G_i , and put e to $E(G_i)$.

Step 2. Construct subgraphs $\tilde{G}_1, \dots, \tilde{G}_m$ deleting all edges in G_i , $1 \leq i \leq m$, which are lighter than w^* . Later we will select w^* so as to retain only heavy edges in subgraphs, though still providing enough edges in each \tilde{G}_i for Step 3.

Step 3. In each subgraph \tilde{G}_i build a Hamiltonian cycle, using a polynomial algorithm, that *with high probability* or *whp* (with probability $\rightarrow 1$ as $n \rightarrow \infty$) finds Hamiltonian cycle in a sparse random graph.

Steps 1 and 2 take $O(n^2)$ time, at Step 3 the chosen algorithm with time complexity $T(n)$ runs m times. So the total time complexity is $O(n^2 + mT(n))$.

Though finding a Hamiltonian cycle is NP-hard in general, the problem can be efficiently solved with small failure probability for suitably randomly selected graphs. In particular, almost all undirected graphs with $1/2(n \log n + n \log \log n + \omega(1))$ edges contain a Hamiltonian cycle [18], whereas for any $\varepsilon > 0$ a graph with less than $(1/2 - \varepsilon)n \log n$ edges whp is not Hamiltonian [9].

At Step 3 of algorithm \tilde{A}_2 we will use algorithm A_{BFF} from [4]. It works for Erdős-Rényi graphs with at least $N = 1/2(n \log n + n \log \log n + c_n)$ edges, has very small failure probability $\delta_{BFF} = O(e^{-2c_n})$, where $c_n = \omega(1)$, and the running-time $O(n^{3+o(1)})$. This is one of the best algorithms in terms of number of required edges and failure probability. Faster algorithms like [2] with $O(n \log^2 n)$ running-time usually have much worse δ and N . The algorithms from [2,4] exploit similar ideas: they grow the partial path from its one or both ends, adding a new edge that leads outside the constructed path or rotating the path, and run until they meet some restriction on the number of allowed steps or the number of allowed rotations.

3.1 Probabilistic Analysis of the Algorithm \tilde{A}_2

Proposition 1. [2] For all n, p, β , with n integer, $0 \leq p \leq 1$, $0 \leq \beta \leq 1$

$$\sum_{k=0}^{\lfloor (1-\beta)np \rfloor} \binom{n}{k} p^k (1-p)^{n-k} \leq \exp\{-\beta^2 np/2\}.$$

Theorem 4. Let the weights of the input graph be i.i.d. random reals with a uniform distribution $UNI[a, b]$, $0 \leq a \leq b$. Using algorithm A_{BFF} [4] at Step 3, for $m = O(n^{1-\theta})$, $0 < \theta < 1$, algorithm \tilde{A}_2 is asymptotically optimal with the following performance guarantees:

$$\varepsilon_{\tilde{A}_2} = O(\ln n / n^\theta), \quad \delta_{\tilde{A}_2} = O(e^{-n+(1-\theta)\log n}).$$

Proof. Recall that Erdős-Rényi random graph is a graph, where each edge is present with probability p , independent from every other edge. Another definition

of Erdős-Rényi random graph is that it is a graph chosen uniformly and independently from a collection of all graphs with n vertices and N edges.

At Step 1 we create random graphs G_1, \dots, G_m , where each edge is present with probability $1/m$, independently of other edges. At Step 2 we delete edges that are lighter than w^* from G_1, \dots, G_m . Thus, $\tilde{G}_1, \dots, \tilde{G}_m$ are random Erdős-Rényi graphs, where each edge exists with probability

$$p = \frac{1 - \text{UNI}(w^*)}{m} = \frac{b - w^*}{(b - a)m}, \quad (4)$$

independently of other edges.

For the algorithm A_{BBF} at Step 3 to succeed whp, each subgraph $\tilde{G}_1, \dots, \tilde{G}_m$ should contain at least $1/2(n \log n + n \log \log n + c_n)$ edges. Set $c_n = n \log(n/\log n)$. Let δ' be the probability that there are less than $n \log n$ edges in \tilde{G}_i , $1 \leq i \leq m$, at the beginning of Step 3. Using Proposition 1 we have:

$$\delta' = \sum_{j=0}^{n \log n - 1} \binom{\frac{n(n-1)}{2}}{j} p^j (1-p)^{\frac{n(n-1)}{2} - j} \leq \exp\left(-\frac{4(\log n + 1)}{n-1} pn\right) \leq e^{-n},$$

if $p \geq \frac{4(\log n + 1)}{n-1}$. Combining it with (4), we get the following lower bound w^* on the weight of edges left in subgraphs:

$$w^* = b - \frac{4m(\log n + 1)}{n-1} (b-a). \quad (5)$$

The optimum weight of the MAX m -PSP solution in our case is $OPT \leq mnb$. If the algorithm A_{BFF} at Step 3 does not fail for each subgraph \tilde{G}_i , $1 \leq i \leq m$, then the weight of the obtained approximate solution is $F_A > w^*mn$. So if $m = O(n^{1-\theta})$, $0 < \theta < 1$, for the relative error we have:

$$\varepsilon_{\tilde{A}_2} = \frac{OPT - F_A}{OPT} < 1 - \frac{w^*mn}{mnb} \leq \frac{4m(\log n + 1)}{n-1} = O\left(\frac{\ln n}{n^\theta}\right).$$

Finally, the failure probability $\delta_{\tilde{A}_2}$ of the algorithm is at most the union bound of the probabilities that the Step 3 fails for subgraphs \tilde{G}_i . This happens with probability δ' , if there were not enough edges in \tilde{G}_i , and with probability $\delta_{BFF} = O(e^{-2n \log(n/\log n)})$ the algorithm A_{BFF} at Step 3 fails on \tilde{G}_i . Thus, if $m = O(n^{1-\theta})$, where $0 < \theta < 1$, we have:

$$\delta_{\tilde{A}_2} \leq m(\delta' + \delta_{BFF}) \leq m(e^{-n} + O(e^{-2n \log(n/\log n)})) = O(e^{-n+(1-\theta)\log n}). \quad \square$$

4 Conclusion

In this paper we propose two algorithms \tilde{A}_1 and \tilde{A}_2 solving the MAX m -PSP with different and identical weight functions, respectively. The algorithms are simple and run in polynomial time. Algorithm \tilde{A}_1 has $O(mn^2)$ running-time

which is linear, considering that the input is an $(m \times n \times n)$ matrix. The time complexity of the algorithm \tilde{A}_2 is $O(mn^{3+o(1)})$, but it can be improved up to $O(n^2 + mn \log^2 n)$, if at Step 3 we use the algorithm from [2], which is faster but has worse failure probability. For both algorithms we have carried out the probabilistic analysis. Assuming that the given weights of edges are i.i.d. random reals with uniform distribution on $[a, b]$, we proved that the algorithms are asymptotically optimal for $m = o(n)$. In contrast to the minimum problem, the asymptotic optimality conditions do not depend on the boundaries of the interval $[a, b]$, $0 \leq a \leq b$. Since the algorithm \tilde{A}_2 is less sophisticated than the algorithm \tilde{A}_1 , it has worse relative error, that nevertheless tends to zero as the number of vertices n grows. On the other hand, the algorithm \tilde{A}_2 is suitable for solving both problems with different and identical weight functions, whereas for the algorithm \tilde{A}_1 the proven performance guarantees are obtained only in the case of different weight functions.

It is also worth noting that the obtained results can be extended to a larger class of instances.

Corollary 1. *For both algorithms \tilde{A}_1 and \tilde{A}_2 the obtained performance guarantees are also valid in the case of a distribution function of inputs $\hat{f}(x)$ dominated by the uniform distribution $UNI(x)$: $\hat{f}(x) \leq UNI(x)$ for all $x \in (-\infty, +\infty)$.*

Acknowledgments. The authors are supported by the Russian Foundation for Basic Research grants 16-31-00389 and 15-01-00976, Russian Ministry of Science and Education under 5-100 Excellence Program, and the grant of Presidium of RAS (program 8, project 227).

References

1. Ageev, A.A., Baburin, A.E., Gimadi, E.K.: A $3/4$ approximation algorithms for finding two disjoint Hamiltonian cycles of maximum weight. *J. Appl. Indust. Math.* **1**(2), 142–147 (2007)
2. Angluin, D., Valiant, L.G.: Fast probabilistic algorithms for Hamiltonian circuits and matchings. *J. Comp. Syst. Sci.* **18**(2), 155–193 (1979)
3. Baburin, A.E., Gimadi, E.K.: On the asymptotic optimality of an algorithm for solving the maximum m -PSP in a multidimensional euclidean space. *Proc. Steklov Inst. Math.* **272**(1), 1–13 (2011)
4. Bollobás, B., Fenner, T.I., Frieze, A.M.: An algorithm for finding Hamilton paths and cycles in random graphs. *Combinatorica* **7**, 327–341 (1987)
5. Climer, S., Zhang, W.: Rearrangement clustering: pitfalls, remedies, and applications. *JMLR* **7**, 919–943 (2006)
6. De Kort, J.B.J.M.: Upper bounds and lower bounds for the symmetric K-Peripatetic Salesman Problem. *Optimization* **23**(4), 357–367 (1992)
7. De Kort, J.B.J.M.: A branch and bound algorithm for symmetric 2-Peripatetic Salesman Problems. *Eur. J. Oper. Res.* **70**, 229–243 (1993)
8. Duchenne, E., Laporte, G., Semet, F.: The undirected m-Peripatetic Salesman Problem: polyhedral results and new algorithms. *J. Oper. Res.* **55**(5), 949–965 (2007)

9. Erdős, P., Rényi, A.: On random graphs I. *Publ. Math. Debrecen* **6**, 290–297 (1959)
10. Gimadi, E.K., Glazkov, Y.V., Tsidulko, O.Y.: The probabilistic analysis of an algorithm for solving the m-planar 3-dimensional assignment problem on one-cycle permutations. *J. Appl. Ind. Math.* **8**(2), 208–217 (2014)
11. Gimadi, E.K., Istomin, A.M., Tsidulko, O.Y.: On asymptotically optimal approach to the m-Peripatetic Salesman Problem on random inputs. In: Kochetov, Y., Khachay, M., Beresnev, V., Nurminski, E., Pardalos, P. (eds.) DOOR 2016. LNCS, vol. 9869, pp. 136–147. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44914-2_11
12. Gimadi, E.K., Ivonina, E.V.: Approximation algorithms for the maximum 2-Peripatetic Salesman Problem. *J. Appl. Ind. Math.* **6**(3), 295–305 (2012)
13. Glebov, A.N., Gordeeva, A.V.: An algorithm with approximation ratio 5/6 for the metric maximum m -PSP. In: Kochetov, Y., Khachay, M., Beresnev, V., Nurminski, E., Pardalos, P. (eds.) DOOR 2016. LNCS, vol. 9869, pp. 159–170. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44914-2_13
14. Glebov, A.N., Zambalaeva, D.Z.: A polynomial algorithm with approximation ratio 7/9 for the maximum two Peripatetic Salesmen Problem. *J. Appl. Ind. Math.* **6**(1), 69–89 (2012)
15. Johnson, D.S., Krishnan, S., Chhugani, J., Kumar, S., Venkatasubramanian, S.: Compressing large boolean matrices using reordering techniques. In: 30th International Conference on Very Large Databases (VLDB), pp. 13–23 (2004)
16. Johnson, O., Liu, J.: A traveling salesman approach for predicting protein functions. *Source Code Biol. Med.* **1**(3), 9–16 (2006)
17. Kaplan, H., Lewenstein, M., Shafir, N., Sviridenko, M.: Approximation algorithms for asymmetric TSP by decomposing directed regular multigraphs. *J. ACM* **52**(4), 602–626 (2005)
18. Komlos, J., Szemerédi, E.: Limit distributions for the existence of Hamilton circuits in a random graph. *Discrete Math.* **43**, 55–63 (1983)
19. Krarup, J.: The Peripatetic Salesman and some related unsolved problems. In: Combinatorial Programming, Methods and Applications, pp. 173–178. Reidel, Dordrecht (1975)
20. Petrov, V.V.: Limit Theorems of Probability Theory. Sequences of Independent Random Variables. Clarendon Press, Oxford (1995)
21. Ray, S.S., Bandyopadhyay, S., Pal, S.K.: Gene ordering in partitive clustering using microarray expressions. *J. Biosci.* **32**(5), 1019–1025 (2007)
22. Song, L., Zhang, Yu., Peng, X., Wang, Z., Gildea, D.: AMR-to-text generation as a Traveling Salesman Problem. In: Proceedings of 2016 Conference on Empirical Methods in Natural Language Processing (2016)

A Randomized Algorithm for 2-Partition of a Sequence

Alexander Kel'manov^{1,2(✉)}, Sergey Khamidullin¹,
and Vladimir Khandeев^{1,2(✉)}

¹ Sobolev Institute of Mathematics, Novosibirsk, Russia
{kelm,kham,khandedev}@math.nsc.ru

² Novosibirsk State University, Novosibirsk, Russia

Abstract. In the paper we consider one strongly NP-hard problem of partitioning a finite Euclidean sequence into two clusters minimizing the sum over both clusters of introcluster sum of squared distances from clusters elements to their centers. The cardinalities of clusters are assumed to be given. The center of the first cluster is unknown and is defined as the mean value of all points in the cluster. The center of the second one is the origin. Additionally, the difference between the indexes of two consequent points from the first cluster is bounded from below and above by some constants. A randomized algorithm for the problem is proposed. For an established parameter value, given a relative error $\varepsilon > 0$ and fixed $\gamma \in (0, 1)$, this algorithm allows to find a $(1 + \varepsilon)$ -approximate solution of the problem with a probability of at least $1 - \gamma$ in polynomial time. The conditions are established under which the algorithm is polynomial and asymptotically exact.

Keywords: Partitioning · Sequence · Euclidean space
Minimum sum-of-squared distances · NP-hardness
Randomized algorithm · Asymptotic accuracy

1 Introduction

The subject of this study is a strongly NP-hard problem of partitioning a finite sequence of points of Euclidean space into two clusters. The goal of the study is to substantiate a randomized algorithm for its solution.

This study is motivated by the lack of studies on the problem and its relevance, in particular, to clustering and analysis of sequences (time series) and also to natural science and technical applications in which one needs to classify the time-ordered data from numerical experiments or results of monitoring of states of some objects (see, e.g., [1–6], the references therein, and the next section).

Other motivations for the study are the following two facts: (1) the absence of randomized algorithms for the considered problem (existing algorithms are characterized in the following section), (2) a well known [7] property of randomized algorithms — in many cases, they take less time than algorithms of other

types to find an efficient approximate solution with guaranteed accuracy and failure probability. The last fact is especially important for the solution of the Big data problem.

This is the incremental work to the results previously obtained in [8–11] for the considered problem. In fact, below we propose the first randomized algorithm for this strongly NP-hard problem.

The paper has the following structure. The next section contains the problem formulation and its treatment. In Sect. 3, known results are presented and our new result is announced. In Sect. 4 we formulate and prove some basic properties exploited by our algorithm. A randomized algorithm is presented in Sect. 5. Finally, we give some directions for future researches.

2 Problem Formulation, Its Treatment, and Related Problems

Everywhere below \mathbb{R} denotes the set of real numbers, $\|\cdot\|$ denotes the Euclidean norm, and $\langle \cdot, \cdot \rangle$ denotes the scalar product.

Formally, we consider the following problem (see also [8–11]).

Problem 1 (Minimum Sum-of-Squares 2-Clustering problem on sequence with given center of one cluster and cluster cardinalities). Given a sequence $\mathcal{Y} = (y_1, \dots, y_N)$ of points from \mathbb{R}^q and positive integers T_{\min} , T_{\max} and $M > 1$. Find a subset $\mathcal{M} = \{n_1, \dots, n_M\} \subseteq \mathcal{N} = \{1, \dots, N\}$ of indexes of elements in sequence \mathcal{Y} that minimizes the objective function

$$F(\mathcal{M}) = \sum_{j \in \mathcal{M}} \|y_j - \bar{y}(\mathcal{M})\|^2 + \sum_{j \in \mathcal{N} \setminus \mathcal{M}} \|y_j\|^2,$$

where $\bar{y}(\mathcal{M}) = \frac{1}{|\mathcal{M}|} \sum_{j \in \mathcal{M}} y_j$ is the centroid (the geometric center) of set $\{y_j \mid j \in \mathcal{M}\}$, provided that

$$T_{\min} \leq n_m - n_{m-1} \leq T_{\max} \leq N, \quad m = 2, \dots, M, \quad (1)$$

for the elements of (n_1, \dots, n_M) .

For readers interested in applied aspects we recall one of the treatments of the problem (see [8–14]). Given a sequence \mathcal{Y} containing N time-ordered results y_1, \dots, y_N of measurements of the tuple y of q numerical characteristics of some object that can be in two states, active and passive. In the passive state all elements of the tuple equal to zero, while in the active state at least one element is nonzero. In addition, the positive integers T_{\min} and T_{\max} are given, which correspond to minimal and maximal time intervals between two consecutive active states of the object. The correspondence of the sequence elements to some state of the object is unknown. The problem is to divide the sequence into the two clusters (subsequences) corresponding to the passive and active states of the object and estimate the set of characteristics of the object in the active state.

Formalization of this simple meaningful problem, in particular, in the form of an approximation problem, induces (see [12–14]) Problem 1. The approximation problem is to find an approximating sequence for \mathcal{Y} having the following structure

$$\dots 0x0 \dots 0x0 \dots 0x0 \dots$$

under the criterion of minimum sum-of-squared deviations. Here $x \in \mathbb{R}^q$ is an unknown point corresponding to the active state, 0 is the origin corresponding to the passive state, and the number of zero points is unknown and defined by the constraints (1). Herein the optimal approximating sequence has the following form

$$\dots 0\bar{y}(\mathcal{M})0 \dots 0\bar{y}(\mathcal{M})0 \dots 0\bar{y}(\mathcal{M})0 \dots$$

In this sequence $\bar{y}(\mathcal{M})$ is a centroid of the first cluster $\{y_j \mid j \in \mathcal{M}\}$, where \mathcal{M} is determined as the result of solving Problem 1. The centroid is an estimate for the point x . The multisubset $\{y_j \mid j \in \mathcal{M}\}$ of the sequence \mathcal{Y} corresponds to the active state of the object.

For the problem's statistical treatment related to noise-proof analysis of time series, which also induces Problem 1, the reader is referred to [12–14].

3 Known and Obtained Results

Recall the known results for Problem 1. First, note that a special case of Problem 1 where $T_{\min} = 1$ and $T_{\max} = N$ is equivalent [8] to the strongly NP-hard problem of partitioning a set [15], which does not admit [16] FPTAS unless P = NP. In other words, Problem 1 of partitioning a sequence is the generalization of the strongly NP-hard problem of partitioning a set. Therefore, according to [17], Problem 1 also admits neither exact polynomial, nor exact pseudopolynomial, nor FPTAS unless P = NP.

In [8], the variant of Problem 1 in which T_{\min} and T_{\max} are the parameters was analyzed. In the cited work it was shown that Problem 1 is strongly NP-hard for any $T_{\min} < T_{\max}$. In the trivial case when $T_{\min} = T_{\max}$, the problem is solvable in polynomial time.

A 2-approximation algorithm for Problem 1 with $\mathcal{O}(N^2(MN + q))$ running time was presented in [9].

Special cases of the problem were studied in [10, 11]. In [10], for the case of integer inputs and fixed space dimension q , an exact pseudopolynomial algorithm was proposed. The running time of the algorithm is $\mathcal{O}(N^3(MD)^q)$, where D is the maximal absolute value of the coordinates of input points. For the case with fixed space dimension in [11] an FPTAS was constructed which, given a relative error ε , finds a $(1 + \varepsilon)$ -approximate solution of Problem 1 in $\mathcal{O}(MN^3(1/\varepsilon)^{q/2})$ time.

The main result of this paper is a randomized algorithm for Problem 1. For an established parameter value, given a relative error $\varepsilon > 0$ and fixed $\gamma \in (0, 1)$, this algorithm allows to find a $(1 + \varepsilon)$ -approximate solution of the problem with a probability of at least $1 - \gamma$ in $\mathcal{O}(qMN^2)$ time. The conditions are established

under which the algorithm is asymptotically exact and its time complexity is $\mathcal{O}(qMN^3)$.

4 The Basics of the Algorithm

To construct the algorithm, we need several basic assertions, an auxiliary problem and an exact polynomial algorithm for its solution.

The probabilistic base of the algorithm is the following lemma (see [18]).

Lemma 1. *Let \mathcal{Z} be an arbitrary set of points from \mathbb{R}^q of cardinality N , $\mathcal{C} \subseteq \mathcal{Z}$ with $|\mathcal{C}| = M$, and \mathcal{T} be a multiset obtained by randomly and independently choosing k elements from \mathcal{Z} with replacement. Additionally, let $\bar{z}(\mathcal{C}) = \frac{1}{M} \sum_{z \in \mathcal{C}} z$ and $\bar{z}(\mathcal{T} \cap \mathcal{C}) = \frac{1}{|\mathcal{T} \cap \mathcal{C}|} \sum_{z \in \mathcal{T} \cap \mathcal{C}} z$ be the centroids of set \mathcal{C} and multiset $\mathcal{T} \cap \mathcal{C}$, respectively. Then, for any positive integer $t \leq k$ and arbitrary $\delta \in (0, 1)$,*

$$\Pr \left(\sum_{z \in \mathcal{C}} \|z - \bar{z}(\mathcal{T} \cap \mathcal{C})\|^2 \geq \left(1 + \frac{1}{\delta t}\right) \sum_{z \in \mathcal{C}} \|z - \bar{z}(\mathcal{C})\|^2 \mid |\mathcal{T} \cap \mathcal{C}| \geq t \right) \leq \delta.$$

The geometrical foundations of the algorithm are given by the following lemma (see [16]).

Lemma 2. *Let*

$$S(\mathcal{M}, x) = \sum_{n \in \mathcal{M}} \|y_n - x\|^2 + \sum_{n \in \mathcal{N} \setminus \mathcal{M}} \|y_n\|^2, \quad x \in \mathbb{R}^q, \quad \mathcal{M} \subseteq \mathcal{N}, \quad (2)$$

where elements of the set $\mathcal{M} = \{n_1, \dots, n_M\}$ satisfy the constraints (1). Then the following statements are true:

- (1) for any fixed subset $\mathcal{M} \subseteq \mathcal{N}$ the minimum of function (2) over $x \in \mathbb{R}^q$ is reached at the point $x = \bar{y}(\mathcal{M})$, and is equal to $F(\mathcal{M})$;
- (2) for any fixed point $x \in \mathbb{R}^q$ the minimum of function

$$S^x(\mathcal{M}) = \sum_{n \in \mathcal{M}} \|y_n - x\|^2 + \sum_{n \in \mathcal{N} \setminus \mathcal{M}} \|y_n\|^2, \quad \mathcal{M} \subseteq \mathcal{N},$$

over set \mathcal{M} of fixed cardinality M is reached at the subset \mathcal{M}^x of the element indexes of \mathcal{Y} that maximize function

$$G^x(\mathcal{M}) = \sum_{n \in \mathcal{M}} \langle y_n, x \rangle, \quad \mathcal{M} \subseteq \mathcal{N}. \quad (3)$$

The computational basis of our algorithm is an exact polynomial-time algorithm for solving the following auxiliary problem.

Problem 2. Given a sequence $\mathcal{Y} = (y_1, \dots, y_N)$ of points from \mathbb{R}^q , a point $x \in \mathbb{R}^q$, and some positive integers T_{\min} , T_{\max} , and $M > 1$. Find a subset $\mathcal{M} = \{n_1, \dots, n_M\} \subseteq \mathcal{N}$ of the element indexes of \mathcal{Y} that maximize (3) under constraints (1) on the members of (n_1, \dots, n_M) .

To explain the algorithm for solving this auxiliary problem, we define the function

$$g^x(n) = \langle y_n, x \rangle, \quad n \in \mathcal{N}, \quad (4)$$

where $x \in \mathbb{R}^q$ is an arbitrary fixed point, and y_n is the n th element of the input sequence \mathcal{Y} .

In accordance with the definition (4), for the objective function (3) we have

$$G^x(\mathcal{M}) = \sum_{n \in \mathcal{M}} g^x(n), \quad \mathcal{M} \subseteq \mathcal{N},$$

where the members of $\mathcal{M} = \{n_1, \dots, n_M\}$ satisfy constraints (1), and, in accordance with the second statement of Lemma 2, the following equality holds

$$\mathcal{M}^x = \arg \min_{\mathcal{M}} S^x(\mathcal{M}) = \arg \max_{\mathcal{M}} G^x(\mathcal{M}).$$

The next lemma and its corollary contain the dynamic programming scheme which finds an optimal solution \mathcal{M}^x of Problem 2. The scheme is based on the results of [9, 19].

Lemma 3. *For any positive integer $M > 1$ such that $(M - 1)T_{\min} \leq N - 1$, and an arbitrary point $x \in \mathbb{R}^q$, the optimal value $G_{\max}^x = \max_{\mathcal{M}} G^x(\mathcal{M})$ of the objective function of Problem 2 can be found as*

$$G_{\max}^x = \max_{n \in \omega_M} G_M^x(n), \quad (5)$$

and the values $G_M^x(n)$, $n \in \omega_M$, are calculated by the recurrent formulas

$$G_m^x(n) = \begin{cases} g^x(n), & \text{for } n \in \omega_1, m = 1; \\ g^x(n) + \max_{j \in \gamma_{m-1}^-(n)} G_{m-1}^x(j), & \text{for } n \in \omega_m, m = 2, \dots, M, \end{cases} \quad (6)$$

where ω_m and $\gamma_{m-1}^-(n)$ are given as

$$\omega_m = \{n \mid 1 + (m - 1)T_{\min} \leq n \leq N - (M - m)T_{\min}\}, \quad m = 1, \dots, M,$$

$$\gamma_{m-1}^-(n) = \{j \mid \max\{1 + (m - 2)T_{\min}, n - T_{\max}\} \leq j \leq n - T_{\min}\}, \\ n \in \omega_m, \quad m = 2, \dots, M.$$

Corollary 1. *The elements n_1^x, \dots, n_M^x of an optimal set \mathcal{M}^x can be found by the following recurrent formulas:*

$$n_M^x = \arg \max_{n \in \omega_M} G_M^x(n), \quad (7)$$

$$n_{m-1}^x = \arg \max_{n \in \gamma_m^-(n_m^x)} G_m^x(n), \quad m = M, M - 1, \dots, 2. \quad (8)$$

The algorithm that implements this scheme can be written in the following form.

Algorithm \mathcal{A}_1

Input: sequence \mathcal{Y} , point $x \in \mathbb{R}^q$, positive integers T_{\min} , T_{\max} , and M .

Step 1. Compute the values $g^x(n)$ for $n \in \mathcal{N}$ using Formula (4).

Step 2. Using Formula (6), compute the values $G_m^x(n)$ for each $n \in \omega_m$ and $m = 1, \dots, M$.

Step 3. Find the maximum G_{\max}^x of the objective function G^x by Formula (5), and the optimal set $\mathcal{M}^x = \{n_1^x, \dots, n_M^x\}$ by Formulae (7), (8).

Output: the set \mathcal{M}^x .

In [9, 19], it was shown that Algorithm \mathcal{A}_1 finds the optimal solution of Problem 2 in $\mathcal{O}(N(M(T_{\max} - T_{\min} + 1) + q))$ time. At that the value $T_{\max} - T_{\min} + 1$ is at most N . Therefore, the algorithm running time can be estimated as $\mathcal{O}(N(MN + q))$.

5 Randomized Algorithm

The idea of the proposed approach to Problem 1 is as follows. A finite multiset is formed by random and independent choice (with replacement) of elements from the input sequence. For the centroid of each nonempty multisubset of this multiset, using dynamic programming scheme, we solve Problem 2 of maximizing the auxiliary objective function. As a result of solving this problem, we find a feasible set of element indexes. This set is treated as a candidate for the solution of the initial problem and is included in the family of solution candidates. From the obtained family as the final solution we choose the set for which the objective function value of Problem 1 is minimal.

Let us formulate an algorithm for Problem 1 that implements the described approach.

Algorithm \mathcal{A}

Input: sequence \mathcal{Y} , positive integers T_{\min} , T_{\max} , and M , and positive integer parameter k .

Step 1. Generate a multiset \mathcal{T} by independently and randomly choosing k elements one after another (with replacement) from \mathcal{Y} .

Step 2. For each nonempty multisubset \mathcal{H} of \mathcal{T} , compute the centroid $\bar{y}(\mathcal{H})$ and, using Algorithm \mathcal{A}_1 , construct an optimal solution $\mathcal{M}^{\bar{y}(\mathcal{H})}$ of Problem 2 (for $x = \bar{y}(\mathcal{H})$).

Step 3. In the family of sets found at Step 2 choose $\mathcal{M}^{\bar{y}(\mathcal{H})}$ minimizing the value $F(\mathcal{M}^{\bar{y}(\mathcal{H})})$ as a solution $\mathcal{M}_{\mathcal{A}}$ of the problem. If there are several optimal values, then choose any of them.

Output: the set $\mathcal{M}_{\mathcal{A}}$.

Theorem 1. *For arbitrary $\delta \in (0, 1)$ and positive integer $t \leq k$, Algorithm \mathcal{A} finds a $(1 + \frac{1}{\delta t})$ -approximate solution of Problem 1 in $\mathcal{O}(2^k(qk + N(M(T_{\max} - T_{\min} + 1) + q)))$ time with a probability of at least $1 - (\delta + \alpha)$, where $\alpha = \sum_{i=0}^{t-1} \binom{k}{i} \left(\frac{M}{N}\right)^i \left(1 - \frac{M}{N}\right)^{k-i}$.*

Proof. Let \mathcal{M}^* be the optimal solution of Problem 1, and let \mathcal{T}' be the multiset of element indexes of \mathcal{T} , that is, $\mathcal{T}' = \{y_i \mid i \in \mathcal{T}'\}$.

Assume that, at Step 1, the multiset \mathcal{T} is chosen so that $|\mathcal{T}' \cap \mathcal{M}^*| \geq 1$. Then, obviously, the multiset $\mathcal{H} = \{y_i \mid i \in \mathcal{T}' \cap \mathcal{M}^*\}$ was considered at Step 2. Let \mathcal{M}' be the optimal solution of Problem 2 (for $x = \bar{y}(\mathcal{H})$), which was constructed at the same step.

By the definition of Step 3,

$$F(\mathcal{M}_A) \leq F(\mathcal{M}'). \quad (9)$$

Additionally, from the first statement of Lemma 2 we have

$$F(\mathcal{M}') = S^{\bar{y}(\mathcal{M}')}(\mathcal{M}') \leq S^{\bar{y}(\mathcal{H})}(\mathcal{M}'). \quad (10)$$

Furthermore, since Algorithm \mathcal{A}_1 finds an optimal solution of Problem 2 of maximization of (3), from the second statement of Lemma 2 we have the inequality

$$S^{\bar{y}(\mathcal{H})}(\mathcal{M}') \leq S^{\bar{y}(\mathcal{H})}(\mathcal{M}^*). \quad (11)$$

Combining (9)–(11), we conclude that, for $|\mathcal{T}' \cap \mathcal{M}^*| \geq 1$, it holds that

$$\begin{aligned} F(\mathcal{M}_A) &\leq F(\mathcal{M}') \leq S^{\bar{y}(\mathcal{H})}(\mathcal{M}') \\ &\leq S^{\bar{y}(\mathcal{H})}(\mathcal{M}^*) = \sum_{j \in \mathcal{M}^*} \|y_j - \bar{y}(\mathcal{H})\|^2 + \sum_{j \in \mathcal{N} \setminus \mathcal{M}^*} \|y_j\|^2. \end{aligned} \quad (12)$$

It is easy to show that Lemma 1 can be applied for the indexes sets. In this way, we have that, for $|\mathcal{T}' \cap \mathcal{M}^*| \geq t$, with a probability of at least $1 - \delta$, the following holds

$$\sum_{j \in \mathcal{M}^*} \|y_j - \bar{y}(\mathcal{H})\|^2 < \left(1 + \frac{1}{\delta t}\right) \sum_{j \in \mathcal{M}^*} \|y_j - \bar{y}(\mathcal{M}^*)\|^2. \quad (13)$$

Finally, combining (12) and (13), we conclude that, for $|\mathcal{T}' \cap \mathcal{M}^*| \geq t$, with a probability of at least $1 - \delta$,

$$\begin{aligned} F(\mathcal{M}_A) &\leq \sum_{j \in \mathcal{M}^*} \|y_j - \bar{y}(\mathcal{H})\|^2 + \sum_{j \in \mathcal{N} \setminus \mathcal{M}^*} \|y_j\|^2 \\ &< \left(1 + \frac{1}{\delta t}\right) \sum_{j \in \mathcal{M}^*} \|y_j - \bar{y}(\mathcal{M}^*)\|^2 + \sum_{j \in \mathcal{N} \setminus \mathcal{M}^*} \|y_j\|^2 \\ &\leq \left(1 + \frac{1}{\delta t}\right) \left(\sum_{j \in \mathcal{M}^*} \|y_j - \bar{y}(\mathcal{M}^*)\|^2 + \sum_{j \in \mathcal{N} \setminus \mathcal{M}^*} \|y_j\|^2 \right) = \left(1 + \frac{1}{\delta t}\right) F(\mathcal{M}^*). \end{aligned}$$

Consequently,

$$\Pr \left(F(\mathcal{M}_A) < \left(1 + \frac{1}{\delta t}\right) F(\mathcal{M}^*) \mid |\mathcal{T}' \cap \mathcal{M}^*| \geq t \right) > 1 - \delta.$$

Passing from the conditional to unconditional probability yields

$$\Pr \left(F(\mathcal{M}_A) < \left(1 + \frac{1}{\delta t} \right) F(\mathcal{M}^*) \right) > 1 - (\delta + \Pr(|\mathcal{T}' \cap \mathcal{M}^*| < t)),$$

which establishes the probability and accuracy bounds for Algorithm \mathcal{A} , taking into account equality $\Pr(|\mathcal{T}' \cap \mathcal{M}^*| < t) = \sum_{i=0}^{t-1} \binom{k}{i} \left(\frac{M}{N} \right)^i \left(1 - \frac{M}{N} \right)^{k-i}$.

Let us estimate the time complexity of the algorithm. The time complexity of Step 1 is determined by the cardinality of \mathcal{T} and is equal to $\mathcal{O}(k)$. At Step 2, for each multisubset \mathcal{H} of multiset \mathcal{T} , it needs $\mathcal{O}(qk)$ operations to compute the centroid $\bar{y}(\mathcal{H})$ and $\mathcal{O}(N(M(T_{\max} - T_{\min} + 1) + q))$ operations to find the optimal solution of Problem 2. The choice of the least element at Step 3 requires $\mathcal{O}(2^k)$ operations. Summing up the costs required at all steps yields the time complexity bound for the algorithm. \square

Remark 1. The algorithm running time can be estimated as $\mathcal{O}(2^k(qk + N(MN + q)))$, since the value $T_{\max} - T_{\min} + 1$ is at most N .

Note that in accordance with Theorem 1 the value $\frac{1}{\delta t}$ is the relative error of the algorithm, the value $\delta + \alpha$ is the algorithm's failure probability, and the value $1 - (\delta + \alpha)$ is the probability of success of the algorithm.

In the next statement we establish the value of the parameter k as the function of real number β , the relative error ε , and the failure probability γ .

Corollary 2. *Assume that $M \geq \beta N$, where $\beta \in (0, 1)$ is a constant. Then, given $\varepsilon > 0$ and $\gamma \in (0, 1)$ for the fixed parameter $k = \max(\lceil \frac{2}{\beta} \lceil \frac{2}{\gamma \varepsilon} \rceil \rceil, \lceil \frac{8}{\beta} \ln \frac{2}{\gamma} \rceil)$, Algorithm \mathcal{A} finds a $(1 + \varepsilon)$ -approximate solution of Problem 1 with a probability of at least $1 - \gamma$ in $\mathcal{O}(2^k(qk + N(M(T_{\max} - T_{\min} + 1) + q)))$ time.*

We omit the proof of Corollary 2 since it is similar to the proof of the corollary to Theorem 1 in [18]. We note only that in the proof, the value $\alpha = \Pr(|\mathcal{T}' \cap \mathcal{M}^*| < t) = \sum_{i=0}^{t-1} \binom{k}{i} \left(\frac{M}{N} \right)^i \left(1 - \frac{M}{N} \right)^{k-i}$ is bounded by $\frac{\gamma}{2}$ using the Chernoff inequality.

Remark 2. Corollary 2 and Remark 1 imply that for fixed β , γ , and ε the complexity of the algorithm is estimated as $\mathcal{O}(qMN^2)$, since under these conditions the parameter k is also fixed.

Below are the conditions for Algorithm \mathcal{A} to be asymptotically exact.

Theorem 2. *Under the conditions of Theorem 1, let $k = \lceil \log_2 N \rceil$, $\delta = (\log_2 N)^{-1/2}$, $t = \lceil \frac{kM}{2N} \rceil$. Assume that $M \geq \beta N$, where $\beta \in (0, 1)$ is a constant. Then Algorithm \mathcal{A} finds a $(1 + \varepsilon_N)$ -approximate solution of Problem 1 with a probability of at least $1 - \gamma_N$ in $\mathcal{O}(qMN^2(T_{\max} - T_{\min} + 1))$ time, where*

$$\varepsilon_N \leq \frac{2}{\beta} (\log_2 N)^{-1/2} \xrightarrow[N \rightarrow \infty]{} 0,$$

$$\gamma_N \leq (\log_2 N)^{-1/2} + N^{-\frac{\beta}{8 \ln 2}} \xrightarrow[N \rightarrow \infty]{} 0.$$

Proof. The proof of these properties of Algorithm \mathcal{A} of finding a subsequence is similar to the proof of estimates for the algorithm of finding a subset [18].

The time complexity bound of the algorithm follows from the fact that, for $k = \lceil \log_2 N \rceil$,

$$\begin{aligned} 2^k(qk + N(M(T_{\max} - T_{\min} + 1) + q)) \\ = \mathcal{O}(N(q \log_2 N + N(M(T_{\max} - T_{\min} + 1) + q))) \\ = \mathcal{O}(qMN^2(T_{\max} - T_{\min} + 1)). \quad \square \end{aligned}$$

Theorem 2 establishes the conditions under which Algorithm \mathcal{A} is asymptotically exact and, according to Remark 1, its running time is $\mathcal{O}(qMN^3)$.

In Sect. 3 it was noted that in [9], a 2-approximation polynomial-time algorithm was proposed for Problem 1; the running time of the algorithm is $\mathcal{O}(N^2(MN + q))$. It is easy to verify that under the conditions of Remark 2 Algorithm \mathcal{A} is N times faster than the algorithm of [9]. Under the conditions of Theorem 2, Algorithm \mathcal{A} have the same time complexity as the algorithm of [9], but allows to find an asymptotically exact solution instead of 2-approximate solution.

6 Conclusion

In this paper, we proposed a randomized algorithm for a strongly NP-hard problem of partitioning a finite sequence of points of Euclidean space into two clusters. The proposed algorithm allows to find an approximate solution of the problem in polynomial time for the fixed relative error, failure probability, and an established parameter value. The conditions are found under which the algorithm is asymptotically exact and has polynomial time complexity.

In our opinion, the presented technique will be useful for constructing efficient randomized algorithms for other NP-hard problems with close statements that arise, in particular, in natural science and technical applications connected with analysis of time series (signals).

A task of much interest is to construct faster randomized algorithms for the problem and to find special cases of the problem for which construction of linear and sublinear randomized algorithms is possible.

Acknowledgments. This work was supported by the Russian Foundation for Basic Research, project nos. 15-01-00462, 16-31-00186, and 16-07-00168.

References

1. Fu, T.: A review on time series data mining. *Eng. Appl. Artif. Intell.* **24**(1), 164–181 (2011)
2. Kuenzer, C., Dech, S., Wagner, W.: Remote Sensing Time Series. *Remote Sensing and Digital Image Processing*, vol. 22. Springer, Switzerland (2015). <https://doi.org/10.1007/978-3-319-15967-6>
3. Liao, T.W.: Clustering of time series data – a survey. *Pattern Recognit.* **38**(11), 1857–1874 (2005)
4. Aggarwal, C.C.: Data Mining: The Textbook. Springer, Switzerland (2015). <https://doi.org/10.1007/978-3-319-14142-8>
5. Carter, J.A., Agol, E., et al.: Kepler-36: a pair of planets with neighboring orbits and dissimilar densities. *Science* **337**(6094), 556–559 (2012)
6. Kel'manov, A.V., Jeon, B.: A posteriori joint detection and discrimination of pulses in a quasiperiodic pulse train. *IEEE Trans. Sig. Process.* **52**(3), 645–656 (2004)
7. Rajeev, M., Prabhakar, R.: Randomized Algorithms. Cambridge University Press, New York (1995)
8. Kel'manov, A.V., Pyatkin, A.V.: On complexity of some problems of cluster analysis of vector sequences. *J. Appl. Ind. Math.* **7**(3), 363–369 (2013)
9. Kel'manov, A.V., Khamidullin, S.A.: An approximating polynomial algorithm for a sequence partitioning problem. *J. Appl. Ind. Math.* **8**(2), 236–244 (2014)
10. Kel'manov, A.V., Khamidullin, S.A., Khandeev, V.I.: Exact pseudopolynomial algorithm for one sequence partitioning problem. *Autom. Remote Control* **78**(1), 67–74 (2017)
11. Kel'manov, A.V., Khamidullin, S.A., Khandeev, V.I.: A fully polynomial-time approximation scheme for a sequence 2-cluster partitioning problem. *J. Appl. Ind. Math.* **10**(2), 209–219 (2016)
12. Gimadi, E.K., Kel'manov, A.V., Kel'manova, M.A., Khamidullin, S.A.: A posteriori detection of a quasiperiodic fragment with a given number of repetitions in a numerical sequence (in Russian). *Sibirsk. Zh. Ind. Mat.* **9**(1), 55–74 (2006)
13. Gimadi, E.K., Kel'manov, A.V., Kel'manova, M.A., Khamidullin, S.A.: A posteriori detecting a quasiperiodic fragment in a numerical sequence. *Pattern Recognit. Image Anal.* **18**(1), 30–42 (2008)
14. Kel'manov, A.V.: Off-line detection of a quasi-periodically recurring fragment in a numerical sequence. *Proc. Steklov Inst. Math.* **263**(S2), 84–92 (2008)
15. Kel'manov, A.V., Pyatkin, A.V.: Complexity of certain problems of searching for subsets of vectors and cluster analysis. *Comput. Math. Math. Phys.* **49**(11), 1966–1971 (2009)
16. Kel'manov, A.V., Khandeev, V.I.: Fully polynomial-time approximation scheme for a special case of a quadratic euclidean 2-clustering problem. *Comput. Math. Math. Phys.* **56**(2), 334–341 (2016)
17. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
18. Kel'manov, A.V., Khandeev, V.I.: A randomized algorithm for two-cluster partition of a set of vectors. *Comput. Math. Math. Phys.* **55**(2), 330–339 (2015)
19. Kel'manov, A.V., Khamidullin, S.A.: Posterior detection of a given number of identical subsequences in a quasi-periodic sequence. *Comput. Math. Math. Phys.* **41**(5), 762–774 (2001)

An Approximation Scheme for a Weighted Two-Cluster Partition Problem

Alexander Kel'manov^{1,2(✉)}, Anna Motkova^{1,2(✉)}, and Vladimir Shenmaier^{1(✉)}

¹ Sobolev Institute of Mathematics, 4 Koptyug Ave., 630090 Novosibirsk, Russia
`kelm@math.nsc.ru, {anitamo,shenmaier}@mail.ru`

² Novosibirsk State University, 2 Pirogova St., 630090 Novosibirsk, Russia

Abstract. We consider the problem of partitioning a set of Euclidean points into two clusters to minimize the weighted sum of the squared intracluster distances from the elements of the clusters to their centers. The center of one of the clusters is unknown and determined as the average value over all points in the cluster, while the center of the other cluster is the origin. The weight factors for both intracluster sums are given as input. We present an approximation algorithm for the problem, which is based on an adaptive-grid-approach. The algorithm implements a fully polynomial-time approximation scheme (FPTAS) in the case of the fixed space dimension. In the case when the dimension of space is not fixed but is bounded by a slowly growing function of the number of input points, the algorithm realizes a polynomial-time approximation scheme (PTAS).

Keywords: Weighted 2-clustering · NP-hardness · Euclidean space · FPTAS · PTAS

1 Introduction

The subject of this study is a weighted two-cluster partition problem for a finite set of Euclidean points when the center of one cluster is fixed. Our goal is to substantiate an approximation scheme for this problem.

Our research is motivated by insufficient study of the problem into an algorithmic direction and its importance in some applications including, for example, data clustering, pattern recognition, machine learning, statistical problems of joint evaluation and hypotheses testing with heterogeneous samples (see [1–10], papers cited therein and the next section).

This paper develops results of constructing the approximations schemes from [1–3] and has the following structure. Section 2 contains the problem formulation, its interpretation and related problems. In the same section we present known results and announce our new results. In Sect. 3 we formulate and prove some basic geometrical properties exploited in order to substantiate the algorithm. In Sect. 4, our approximation algorithm is presented. Also in Sect. 4 we show that our algorithm is a fully polynomial-time approximation scheme (FPTAS)

when the space dimension is fixed (or bounded by some constant). Finally, in Sect. 5 we present improved algorithm and show that it realizes a polynomial-time approximation scheme (PTAS) when the space dimension is bounded by a slowly growing function of the number of input points.

2 Problem Formulation and Related Problems, Known and New Results

Everywhere below we use the standard notations, namely: \mathbb{R} is the set of the real numbers, \mathbb{R}_+ is the set of positive real numbers, \mathbb{Z} is the set of integers, $\|\cdot\|$ is the Euclidean norm, and $\langle \cdot, \cdot \rangle$ is the scalar product.

We consider the following problem.

Problem 1 (*Weighted variance-based 2-clustering with given center*). Given an N -element set \mathcal{Y} of points from \mathbb{R}^q , a positive integer $M \leq N$, and real numbers (weights) $w_1 > 0$ and $w_2 \geq 0$.

Find a partition of \mathcal{Y} into two clusters \mathcal{C} and $\mathcal{Y} \setminus \mathcal{C}$ minimizing the value of

$$F(\mathcal{C}) = w_1 \sum_{y \in \mathcal{C}} \|y - \bar{y}(\mathcal{C})\|^2 + w_2 \sum_{y \in \mathcal{Y} \setminus \mathcal{C}} \|y\|^2, \quad (1)$$

where $|\mathcal{C}| = M$ and $\bar{y}(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{y \in \mathcal{C}} y$ is the centroid of \mathcal{C} .

It is obvious that we can interpret this problem as a problem of weighted partitioning or clusterization (w_1 and w_2 are the weights). Earlier, the following strongly NP-hard variants of problem under study were explored: $w_1 = 1$ and $w_2 = 0$ (see [1, 4–6, 11–13]), $w_1 = w_2 = 1$ (see [2, 7–9, 14–16]), $w_1 = |\mathcal{C}|$ and $w_2 = N - |\mathcal{C}|$ (see [3, 10, 17, 18]).

The *first variant*, when $w_1 = 1$ and $w_2 = 0$, is known as *M-Variance problem* [13]. This variant is induced (see, for example, [1, 4, 5, 11]) by the problem of searching the subset of similar points. This is one of the easiest problems of data analysis and pattern recognition.

In the *second variant*, when $w_1 = w_2 = 1$, we need to find a partition of the input set into two clusters so as to minimize the sum of two intraccluster sums. The first sum is the sum of squared distances from the elements of the cluster to its centroid. The second one is the sum of squared distances from the elements of the cluster to the origin. This problem is induced by one of the problems of the noise-proof data analysis (see, in particular, [2, 7–9, 14]).

The *third variant* of the problem, when $w_1 = |\mathcal{C}|$ and $w_2 = N - |\mathcal{C}|$, one can interpret as a weighted by the cardinalities two-cluster partition problem. This variant arises, in particularly, in the hypothesis testing problems when the sample is inhomogeneous (see, for example, [3, 10, 17, 18]).

As the noted above variants of the problem, the general weighted problem under study arises, in particular, in Data analysis, Pattern recognition, Machine learning and Data mining (see, for example, [19–23]). Models of the cluster partitioning play the key role in these problems.

Today there are a lot of publications and results for the variants of the problem that could be found in [1–10, 12, 13, 15, 16].

The goal of our research is to generalize the results from [1–3] on the case of an arbitrary $w_1 > 0$ and $w_2 \geq 0$. In the cited papers, approximation schemes that allows to find $(1 + \varepsilon)$ -approximate solution with time complexity $\mathcal{O}\left(qN^2\left(\sqrt{\frac{2q}{\varepsilon}} + 2\right)^q\right)$ were proposed.

In this paper an approximate algorithm for any $w_1 > 0$ and $w_2 \geq 0$ is constructed. It allows to find a $(1 + \varepsilon)$ -approximate solution of the problem for an arbitrary $\varepsilon \in (0, 1)$ in $\mathcal{O}\left(qN^2\left(\sqrt{\frac{2q}{\varepsilon}} + 2\right)^q\right)$ time. It is the same time complexity as time complexity in [1–3].

Moreover, we propose the modification of this algorithm with improved time complexity: $\mathcal{O}\left(\sqrt{q}N^2\left(\frac{\pi e}{2}\right)^{q/2}\left(\sqrt{\frac{2}{\varepsilon}} + 2\right)^q\right)$. The algorithm implements an FPTAS in the case of fixed space dimension and remains polynomial for instances of dimension $\mathcal{O}(\log n)$. In this case it implements a PTAS with $\mathcal{O}\left(N^{C(1.05+\log(2+\sqrt{\frac{2}{\varepsilon}}))}\right)$ time, where C is a positive constant.

3 Basics of the Algorithm

In order to substantiate the algorithms we will need some basic statements. We provide them in this section.

The following two lemmas are well known. Their proofs are presented in many publications (see, for example, [1, 4]).

Lemma 1. *For an arbitrary point $x \in \mathbb{R}^q$, a finite set $\mathcal{Z} \subset \mathbb{R}^q$ and $\bar{z} = \frac{1}{|\mathcal{Z}|} \sum_{z \in \mathcal{Z}} z$ (\bar{z} is the centroid of \mathcal{Z}), it is true that*

$$\sum_{z \in \mathcal{Z}} \|z - x\|^2 = \sum_{z \in \mathcal{Z}} \|z - \bar{z}\|^2 + |\mathcal{Z}| \cdot \|x - \bar{z}\|^2.$$

Lemma 2. *For a finite set $\mathcal{Z} \subset \mathbb{R}^q$, if a point $u \in \mathbb{R}^q$ is closer (in terms of distance) to the centroid \bar{z} of \mathcal{Z} than any point in \mathcal{Z} , then*

$$\sum_{z \in \mathcal{Z}} \|z - u\|^2 \leq 2 \sum_{z \in \mathcal{Z}} \|z - \bar{z}\|^2.$$

Lemma 3. *Let*

$$S(\mathcal{C}, x) = w_1 \sum_{y \in \mathcal{C}} \|y - x\|^2 + w_2 \sum_{y \in \mathcal{Y} \setminus \mathcal{C}} \|y\|^2, \quad \mathcal{C} \subseteq \mathcal{Y}, \quad x \in \mathbb{R}^q.$$

Then the next statements are true:

- (1) *for any nonempty fixed set $\mathcal{C} \subseteq \mathcal{Y}$ the minimum of the function $S(\mathcal{C}, x)$ over $x \in \mathbb{R}^q$ is reached at the point $\bar{y}(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{y \in \mathcal{C}} y$;*

- (2) if $|\mathcal{C}| = M = \text{const}$, then for any fixed point $x \in \mathbb{R}^q$ the minimum of function $S(\mathcal{C}, x)$ over $\mathcal{C} \subseteq \mathcal{Y}$ is reached at the subset \mathcal{B}^x that consists of M points of the set \mathcal{Y} , at which the function

$$g^x(y) = (w_1 - w_2) \|y\|^2 - 2w_1 \langle y, x \rangle, \quad y \in \mathcal{Y}, \quad (2)$$

has the smallest values.

Proof. The first statement follows from Lemma 1 and the definition of the functions S and F . Since $|\mathcal{Y}| = N$ and $|\mathcal{C}| = M$, the second statement follows from the next chain of equalities:

$$\begin{aligned} S(\mathcal{C}, x) &= w_1 \sum_{y \in \mathcal{C}} \|y - x\|^2 + w_2 \sum_{y \in \mathcal{Y} \setminus \mathcal{C}} \|y\|^2 \\ &= w_1 \sum_{y \in \mathcal{C}} \|y\|^2 - 2w_1 \sum_{y \in \mathcal{C}} \langle y, x \rangle + Mw_1 \|x\|^2 + w_2 \left(\sum_{y \in \mathcal{Y}} \|y\|^2 - \sum_{y \in \mathcal{C}} \|y\|^2 \right) \\ &= \sum_{y \in \mathcal{C}} \left\{ (w_1 - w_2) \|y\|^2 - 2w_1 \langle y, x \rangle \right\} + Mw_1 \|x\|^2 + w_2 \sum_{y \in \mathcal{Y}} \|y\|^2. \end{aligned}$$

It remains to note that in the last equality the last two addends do not depend on \mathcal{C} . \square

Lemma 4. Let \mathcal{C}^* be the optimal solution of Problem 1 and let t be the point from the subset \mathcal{C}^* closest to its centroid. Then the following inequality is true

$$\|t - \bar{y}(\mathcal{C}^*)\|^2 \leq \frac{1}{Mw_1} F(\mathcal{B}^t). \quad (3)$$

Proof. By the definition of point t we have

$$\|t - \bar{y}(\mathcal{C}^*)\|^2 \leq \|y - \bar{y}(\mathcal{C}^*)\|^2,$$

where $y \in \mathcal{C}^*$.

Summing up both sides of this inequality over all $y \in \mathcal{C}^*$, we obtain

$$M \|t - \bar{y}(\mathcal{C}^*)\|^2 \leq \sum_{y \in \mathcal{C}^*} \|y - \bar{y}(\mathcal{C}^*)\|^2.$$

Since \mathcal{C}^* is the optimal solution, by using the definition of the function (1) we obtain

$$w_1 \sum_{y \in \mathcal{C}^*} \|y - \bar{y}(\mathcal{C}^*)\|^2 \leq F(\mathcal{C}^*) \leq F(\mathcal{B}^t).$$

That yields the statement of the lemma. \square

Lemma 5. Let

$$\|x - \bar{y}(\mathcal{C}^*)\|^2 \leq \frac{\varepsilon}{2Mw_1} F(\mathcal{B}^t) \quad (4)$$

for an arbitrary $\varepsilon > 0$ and for some $x \in \mathbb{R}^q$. Then the subset \mathcal{B}^x (defined in Lemma 3) is a $(1 + \varepsilon)$ -approximate solution of Problem 1.

Proof. By Lemma 3 it is true that

$$F(\mathcal{B}^t) = S(\mathcal{B}^t, \bar{y}(\mathcal{B}^t)) \leq S(\mathcal{B}^t, t) \leq S(\mathcal{C}^*, t). \quad (5)$$

On the other hand, by Lemma 2

$$\sum_{y \in \mathcal{C}^*} \|y - t\|^2 \leq 2 \sum_{y \in \mathcal{C}^*} \|y - \bar{y}(\mathcal{C}^*)\|^2,$$

so we have

$$S(\mathcal{C}^*, t) \leq 2F(\mathcal{C}^*). \quad (6)$$

Combining (4), (5) and (6) we obtain

$$\|x - \bar{y}(\mathcal{C}^*)\|^2 \leq \frac{\varepsilon}{2Mw_1} F(\mathcal{B}^t) \leq \frac{\varepsilon}{2Mw_1} S(\mathcal{C}^*, t) \leq \frac{\varepsilon}{Mw_1} F(\mathcal{C}^*).$$

Combining Lemmas 1, 3 and (3) we obtain final estimation

$$\begin{aligned} F(\mathcal{B}^x) &= S(\mathcal{B}^x, \bar{y}(\mathcal{B}^x)) \leq S(\mathcal{B}^x, x) \leq S(\mathcal{C}^*, x) \\ &\leq F(\mathcal{C}^*) + Mw_1 \|x - \bar{y}(\mathcal{C}^*)\|^2 \leq (1 + \varepsilon)F(\mathcal{C}^*). \end{aligned}$$
□

4 Approximation Algorithm

In this section, we present the approximation algorithm for Problem 1. Its main idea can be schematically described as follows. For each point of the input set we construct a domain (cube) so that the center of the desired subset necessarily belongs to one of these domains. We generate, using given (as input) the prescribed relative error ε of the solution, a lattice (a grid) that discretizes the cube with a uniform step in all coordinates. For each lattice node, a subset of M points from the input set that have the smallest values of the function (2) is formed. The resulting set is declared as a solution candidate. The candidate that minimizes the objective function is chosen to be the final solution.

This essentially grid approach was used in [1–3] for solving three strongly NP-hard problems that were formulated in the Sect. 1. This paper demonstrates the effectiveness of the grid approach for solving the intractable generalized problem under study.

For an arbitrary point $x \in \mathbb{R}^q$ and positive numbers h and H , we define the set of points

$$\begin{aligned} \mathcal{D}(x, h, H) \\ = \{d \in \mathbb{R}^q \mid d = x + h \cdot (i_1, \dots, i_q), i_k \in \mathbb{Z}, |hi_k| \leq H, k \in \{1, \dots, q\}\} \end{aligned} \quad (7)$$

which is a cubic lattice of size $2H$ centered at the point x with node spacing h .

Remark 1. If for arbitrary points x and $z \in \mathbb{R}^q$ is true $\|z - x\| \leq H$, then the distance from z to the nearest node of the lattice $\mathcal{D}(x, h, H + h/2)$ does not exceed $\frac{h\sqrt{q}}{2}$.

For constructing an algorithmic solution we need to determine adaptively the size H of the lattice and its node spacing h for each point y of the input set \mathcal{Y} so that the domain of the lattice contains the centroid of the desired subset. The node spacing is defined by the relative error ε . To this end we define the functions:

$$H(y) = \sqrt{\frac{1}{Mw_1} F(\mathcal{B}^y)}, \quad y \in \mathcal{Y}, \quad (8)$$

$$h(y, \varepsilon) = \sqrt{\frac{2\varepsilon}{qMw_1} F(\mathcal{B}^y)}, \quad y \in \mathcal{Y}, \quad \varepsilon \in \mathbb{R}_+, \quad (9)$$

where \mathcal{B}^y is the set determined in Lemma 3.

Remark 2. For an arbitrary point $y \in \mathcal{Y}$ the cardinality of the lattice $\mathcal{D}(y, h, H + h/2)$ does not exceed the value

$$L = \left(2 \left\lfloor \frac{H + h/2}{h} \right\rfloor + 1 \right)^q \leq \left(2 \frac{H}{h} + 2 \right)^q = \left(\sqrt{\frac{2q}{\varepsilon}} + 2 \right)^q$$

due to (8) and (9).

The step-by-step description looks like as follows.

Algorithm \mathcal{A}

Input: N -elements set $\mathcal{Y} \subset \mathbb{R}^q$, natural number $M \leq N$, real numbers $w_1 > 0$, $w_2 \geq 0$ and $\varepsilon \in (0, 1)$.

For each point $y \in \mathcal{Y}$ Steps 1–5 are executed.

Step 1. Compute the values $g^y(z)$, $z \in \mathcal{Y}$, using formula (2); find an M -elements subset $\mathcal{B}^y \subseteq \mathcal{Y}$ with the smallest values $g^y(z)$, compute $F(\mathcal{B}^y)$ using formula (1).

Step 2. If $F(\mathcal{B}^y) = 0$, then put $\mathcal{C}_{\mathcal{A}} = \mathcal{B}^y$; exit.

Step 3. Compute H and h using formulae (8) and (9).

Step 4. Construct the lattice $\mathcal{D}(y, h, H + h/2)$ using formula (7).

Step 5. For each node x of the lattice $\mathcal{D}(y, h, H + h/2)$ compute the values $g^x(y)$, $y \in \mathcal{Y}$, using formula (2) and find M -elements subset $\mathcal{B}^x \subseteq \mathcal{Y}$ with the smallest values $g^x(y)$. Compute $F(\mathcal{B}^x)$ using formula (1), remember this value and the set \mathcal{B}^x .

Step 6. In the family $\{\mathcal{B}^x | x \in \mathcal{D}(y, h, H + h/2), y \in \mathcal{Y}\}$ of candidate sets that have been constructed in Steps 1–5, choose as a solution $\mathcal{C}_{\mathcal{A}}$ the set \mathcal{B}^x , for which $F(\mathcal{B}^x)$ is minimal.

Output: the set $\mathcal{C}_{\mathcal{A}}$.

Theorem 1. For any fixed $\varepsilon \in (0, 1)$ Algorithm \mathcal{A} finds a $(1 + \varepsilon)$ -approximate solution of Problem 1 in time:

$$\mathcal{O} \left(qN^2 \left(\sqrt{\frac{2q}{\varepsilon}} + 2 \right)^q \right).$$

Proof. Let us bound the approximation factor of the algorithm. It is obvious, that the algorithm meets the point $t \in \mathcal{Y}$ that is the closest one to the centroid of the optimal subset \mathcal{C}^* while running. By Lemma 4, inequality (3) holds for this point. This inequality and (8) mean that $\|t - \bar{y}(\mathcal{C}^*)\| \leq H(t)$, so the centroid of the optimal subset lies within the cube with the diameter $2H(t)$ and the center in point t .

Let x^* be the node of the grid $\mathcal{D}(t, h, H + h/2)$ that is the nearest to the centroid \mathcal{C}^* . The squared distance from the optimal centroid $\bar{y}(\mathcal{C}^*)$ to the nearest node x^* of the lattice does not exceed $\frac{h^2 q}{4}$ due to Remark 1. It yields the estimate

$$\|x^* - \bar{y}(\mathcal{C}^*)\|^2 \leq \frac{h^2 q}{4} = \frac{\varepsilon}{2Mw_1} F(\mathcal{B}^t). \quad (10)$$

Therefore, the point x^* satisfies the conditions of Lemma 5, and, hence, the set \mathcal{B}^{x^*} is a $(1 + \varepsilon)$ -approximate solution of Problem 1.

Let us evaluate the time complexity of the algorithm. At Step 1 calculation of $g^y(z)$ requires at most $\mathcal{O}(qN)$ -time. Finding the M smallest elements in the set of N elements requires $\mathcal{O}(N)$ operations (for example, using the algorithm of finding the n -th smallest value in an unordered array [24]). Computation of the value $F(\mathcal{B}^y)$ takes $\mathcal{O}(qN)$ time.

Steps 2 and 3 are executed in $\mathcal{O}(q)$ operations. It requires $\mathcal{O}(qL)$ operations for generating the lattice at Step 4 (by Remark 2).

At Step 5, computation of the elements of the set \mathcal{B}^x for each node x of the grid requires $\mathcal{O}(qN)$ time, and the same is true for the computation of $F(\mathcal{B}^x)$ (as computations at Step 1). Thus, at this step the computational time for all nodes of the lattice is $\mathcal{O}(qNL)$.

Since Steps 1–5 are performed N times, the time complexity of these steps is $\mathcal{O}(qN^2L)$. The time complexity of Step 6 is bounded by $\mathcal{O}(NL)$, and the total time complexity of all Steps is $\mathcal{O}(qN^2L) = \mathcal{O}\left(qN^2\left(\sqrt{\frac{2q}{\varepsilon}} + 2\right)^q\right)$. \square

Remark 3. It is clear that in the case of the fixed dimension q of the space Algorithm \mathcal{A} is an FPTAS.

5 Improved Algorithm

We can improve proposed algorithm by an exception of the considerable part of nodes from the process of the calculation.

Indeed, the distance between the centroid of the optimal cluster and one of the points of the input set does not exceed some threshold value H . Hence, it is enough to consider only those nodes of the constructed grids that are located at the distance from their centers that does not exceed H plus a small reserve (defined below in Lemma 6).

For each $y \in \mathcal{Y}$ let $R = H + \frac{h\sqrt{q}}{2}$, where $H = H(y)$, $h = h(y, \varepsilon)$ — the parameters of the grid defined by (8) and (9). Let us construct the reduced spherical lattice

$$\mathcal{D}_R(y, h, H + h/2) = \mathcal{D}(y, h, H + h/2) \cap B(y, R),$$

where $B(y, R) = \{x \in \mathbb{R}^q \mid \|x - y\| \leq R\}$ is the ball of radius R and center y .

Then the next statements are true.

Lemma 6. *For an arbitrary point x of $B(y, H)$, $y \in \mathcal{Y}$, the distance from x to the closest node of the spherical grid $\mathcal{D}_R(y, h, H + h/2)$ does not exceed the value $\frac{h\sqrt{q}}{2}$.*

Proof. Let z be the closest to x node of the cubic lattice $\mathcal{D}(y, h, H + h/2)$. Then due to Remark 1 it is true that $\|z - x\| \leq \frac{h\sqrt{q}}{2}$. On the other hand, $\|x - y\| \leq H$. Combining this fact and the triangle inequality we obtain $\|z - y\| \leq H + \frac{h\sqrt{q}}{2} = R$. Hence, $z \in \mathcal{D}_R(y, h, H + h/2)$. \square

Lemma 7. *For an arbitrary point $y \in \mathcal{Y}$ the cardinality of the lattice $\mathcal{D}_R(y, h, H + h/2)$ does not exceed the value*

$$\frac{1}{\sqrt{\pi q}} \left(\frac{2\pi e}{q} \right)^{q/2} \left(\frac{H}{h} + \sqrt{q} \right)^q.$$

Proof. Since each node z of the lattice $\mathcal{D}_R(y, h, H + h/2)$ lies into the ball $B(y, R)$, by using the triangle inequality we obtain that all points of the q -dimensional cube with the side h and center z lies into the ball $B(y, R + \frac{h\sqrt{q}}{2})$. Hence, the total volume of all this cubes does not exceed the volume of an q -dimensional ball of radius $R + \frac{h\sqrt{q}}{2} = H + h\sqrt{q}$. That yields

$$L_R h^q \leq V_q \left(H + h\sqrt{q} \right)^q,$$

where L_R is the cardinality of the lattice $\mathcal{D}_R(y, h, H + h/2)$ and V_q is the volume of the q -dimensional unit ball that is estimated by well-known formula (see, for example, [25])

$$V_q \leq \frac{1}{\sqrt{\pi q}} \left(\frac{2\pi e}{q} \right)^{q/2}.$$

Combining two last estimations we note the statement of the lemma. \square

Let \mathcal{A}_R be an algorithm that is differ from algorithm \mathcal{A} at Steps 4 and 5. At this steps instead of grids $\mathcal{D}(y, h, H + h/2)$ let us use grids $\mathcal{D}_R(y, h, H + h/2)$. Then the next theorem is true.

Theorem 2. *For any fixed $\varepsilon \in (0, 1)$ Algorithm \mathcal{A}_R finds $(1 + \varepsilon)$ -approximate solution of Problem 1 in*

$$\mathcal{O} \left(\sqrt{q} N^2 \left(\frac{\pi e}{2} \right)^{q/2} \left(\sqrt{\frac{2}{\varepsilon}} + 2 \right)^q \right) \quad (11)$$

time.

Proof. Due to the equality $\frac{H}{h} = \frac{1}{\sqrt{2}}\sqrt{\frac{q}{\varepsilon}}$ and Lemma 7 we obtain

$$L_R \leq \frac{1}{\sqrt{\pi q}} \left(\frac{2\pi e}{q} \right)^{q/2} \left(\frac{1}{\sqrt{2}} \sqrt{\frac{q}{\varepsilon}} + \sqrt{q} \right)^q = \frac{1}{\sqrt{\pi q}} \left(\frac{\pi e}{2} \right)^{q/2} \left(\sqrt{\frac{2}{\varepsilon}} + 2 \right)^q.$$

This fact and the estimations for the time complexity (for Algorithm \mathcal{A}) at Steps 1–6 in Theorem 1 yields the time complexity estimation (11) for Algorithm \mathcal{A}_R .

It remains to note that the approximation factors are similar for both algorithms. Indeed, for some $y \in \mathcal{Y}$ the centroid of the optimal cluster lies in the ball $B(y, H)$ and so, due to Lemma 6, the distance from y to one of the nodes of the reduced lattice $\mathcal{D}_R(y, h, H + h/2)$ does not exceed the value $\frac{h\sqrt{q}}{2}$. It means that the estimation (10) is true, hence, the algorithm finds $(1 + \varepsilon)$ -approximate solution of Problem 1. \square

Remark 4. Due to $\mathcal{D}_R(y, h, H + h/2) \subset \mathcal{D}(y, h, H + h/2)$, the time complexity of Algorithm \mathcal{A}_R is less than time complexity of Algorithm \mathcal{A} for any N, q and ε .

Remark 5. Algorithm \mathcal{A}_R remains polynomial even when the dimension q of the space is bounded by the value $C \log N$, where C is a positive constant. In this case, due to Theorem 2, algorithm finds $(1 + \varepsilon)$ -approximate solution of Problem 1 in $O(N^d \log N)$ time, where $d = \frac{C}{2} \log \frac{\pi e}{2} + C \log(2 + \sqrt{\frac{2}{\varepsilon}}) < C(1.05 + \log(2 + \sqrt{\frac{2}{\varepsilon}}))$. Thus algorithm implements a PTAS in this case.

6 Conclusion

In this paper we presented an approximation algorithm for a quadratic Euclidian problem of weighted partitioning a finite set of point into two clusters with the given center of one cluster. Our algorithm based on an adaptive-grid-approach. It was proved that the algorithm is a fully polynomial-time approximation scheme if the space dimension is bounded by a constant (i.e., if the space dimension is fixed).

The algorithm remains polynomial even in the case when the space dimension is bounded by the value $\mathcal{O}(\log N)$, i.e., in the case when the dimension of space is the slowly growing function of the number of input points. This case is important because the space dimension $\mathcal{O}(\log N)$ is the minimum one when there exist an N -elements set of points with the coordinates from the fixed finite set of values.

In the algorithmical sense, the considered problem is poorly studied. So, it seems important to continue studying the questions on algorithmical approximability of the problem.

Acknowledgments. This work was supported by the Russian Science Foundation (project 16-11-10041).

References

1. Kel'manov, A.V., Romanchenko, S.M.: An FPTAS for a vector subset search problem. *J. Appl. Ind. Math.* **8**(3), 329–336 (2014)
2. Kel'manov, A.V., Khandeev, V.I.: Fully polynomial-time approximation scheme for a special case of a quadratic euclidean 2-clustering problem. *Comput. Math. Math. Phys.* **56**(2), 334–341 (2016)
3. Kel'manov, A., Motkova, A.: A fully polynomial-time approximation scheme for a special case of a balanced 2-clustering problem. In: Kochetov, Y., Khachay, M., Beresnev, V., Nurminski, E., Pardalos, P. (eds.) DOOR 2016. LNCS, vol. 9869, pp. 182–192. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44914-2_15
4. Kel'manov, A.V., Romanchenko, S.M.: An approximation algorithm for solving a problem of search for a vector subset. *J. Appl. Ind. Math.* **6**(1), 90–96 (2012)
5. Kel'manov, A.V., Romanchenko, S.M.: Pseudopolynomial algorithms for certain computationally hard vector subset and cluster analysis problems. *Autom. Remote Control* **73**(2), 349–354 (2012)
6. Shenmaier, V.V.: An approximation scheme for a problem of search for a vector subset. *J. Appl. Ind. Math.* **6**(3), 381–386 (2012)
7. Kel'manov, A.V., Khandeev, V.I.: A randomized algorithm for two-cluster partition of a set of vectors. *Comput. Math. Math. Phys.* **55**(2), 330–339 (2015)
8. Kel'manov, A.V., Khandeev, V.I.: An exact pseudopolynomial algorithm for a problem of the two-cluster partitioning of a set of vectors. *J. Appl. Ind. Math.* **9**(4), 497–502 (2015)
9. Dolgushev, A.V., Kel'manov, A.V., Shenmaier, V.V.: Polynomial-time approximation scheme for a problem of partitioning a finite set into two clusters. *Proc. Steklov Inst. Math.* **295**(1), 47–56 (2016)
10. Kel'manov, A.V., Motkova, A.V.: Exact pseudopolynomial algorithms for a balanced 2-clustering problem. *J. Appl. Ind. Math.* **10**(3), 349–355 (2016)
11. Kel'manov, A.V., Pyatkin, A.V.: NP-completeness of some problems of choosing a vector subset. *J. Appl. Ind. Math.* **5**(3), 352–357 (2011)
12. Shenmaier, V.V.: Solving some vector subset problems by Voronoi diagrams. *J. Appl. Ind. Math.* **10**(4), 560–566 (2016)
13. Aggarwal, A., Imai, H., Katoh, N., Suri, S.: Finding k points with minimum diameter and related problems. *J. Algorithms* **12**(1), 38–56 (1991)
14. Kel'manov, A.V., Pyatkin, A.V.: Complexity of certain problems of searching for subsets of vectors and cluster analysis. *Comput. Math. Math. Phys.* **49**(11), 1966–1971 (2009)
15. Dolgushev, A.V., Kel'manov, A.V.: An approximation algorithm for solving a problem of cluster analysis. *J. Appl. Ind. Math.* **5**(4), 551–558 (2011)
16. Gimadi, E.K., Pyatkin, A.V., Rykov, I.A.: On polynomial solvability of some problems of a vector subset choice in a Euclidean space of fixed dimension. *J. Appl. Ind. Math.* **1**(4), 48–53 (2010)
17. Kel'manov, A.V., Pyatkin, A.V.: NP-hardness of some quadratic euclidean 2-clustering problems. *Dokl. Math.* **92**(2), 634–637 (2015)
18. Kel'manov, A.V., Pyatkin, A.V.: On the complexity of some quadratic euclidean 2-clustering problems. *Comput. Math. Math. Phys.* **56**(3), 491–497 (2016)
19. Aggarwal, C.C.: Data Mining: The Textbook. Springer, Switzerland (2015). <https://doi.org/10.1007/978-3-319-14142-8>
20. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer Science+Business Media, LLC, New York (2006)

21. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York (2009). <https://doi.org/10.1007/978-0-387-84858-7>
22. James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning. Springer Science+Business Media, LLC, New York (2013). <https://doi.org/10.1007/978-1-4614-7138-7>
23. Jain, A.K.: Data clustering: 50 years beyond k -means. Pattern Recognit. Lett. **31**(8), 651–666 (2010)
24. Wirth, N.: Algorithms + Data Structures = Programs. Prentice Hall, New Jersey (1976)
25. Ball, K.: An Elementary Introduction to Modern Convex Geometry. Flavors Geom. **31**, 1–58 (1997). MSRI Publications

Hitting Set Problem for Axis-Parallel Squares Intersecting a Straight Line Is Polynomially Solvable for Any Fixed Range of Square Sizes

Daniel Khachay^{1,2(✉)}, Michael Khachay^{1,2,3}, and Maria Poberiy¹

¹ Krasovsky Institute of Mathematics and Mechanics, Ekaterinburg, Russia
`{dmx, mkhachay}@imm.uran.ru, maschas_briefen@mail.ru`

² Ural Federal University, Ekaterinburg, Russia

³ Omsk State Technical University, Omsk, Russia

Abstract. The Hitting Set Problem is the well known discrete optimization problem adopting interest of numerous scholars in graph theory, computational geometry, operations research, and machine learning. The problem is NP-hard and remains intractable even in very specific settings, e.g., for axis-parallel rectangles on the plane. Recently, for unit squares intersecting a straight line, a polynomial time optimal algorithm was proposed. Unfortunately, the time consumption of this algorithm was $O(n^{145})$. We propose an improved algorithm, whose complexity bound is more than 100 orders of magnitude less. We extend this algorithm to the more general case of the problem and show that the geometric HSP for axis-parallel (not necessarily unit) squares intersected by a line is polynomially solvable for any fixed range of squares to hit. We believe that the obtained theoretical complexity bounds for our algorithms still can be improved further. According to the results of the numerical evaluation presented in the concluding section of the paper, at least for unit squares, an average time consumption bound of our algorithm is less than its deterministic counterpart by 9 orders of magnitude.

Keywords: Hitting Set Problem · Dynamic programming
Computational geometry · Parameterized complexity

1 Introduction

We consider exact algorithms and parameterized complexity of one geometric setting of the famous Hitting Set Problem (HSP), engaging researchers in combinatorial optimization, computational geometry and statistical learning from the early 1980-th.

To the best of our knowledge, HSP gains theoretical interest because it was the first intractable combinatorial optimization problem, whose approximation algorithms were dramatically improved [12] on the basis of Vapnik and Chervonenkis's [17] results in statistical learning theory. The development of randomized algorithms for HSP and related combinatorial problems defined on

range spaces of finite VC-dimension, initiated by seminal papers [1, 6] established a new field in modern computational geometry.

On the other hand, the concepts of hitting set and classifier ensemble, making decisions by some voting logic, seem to be related very closely. Consequently, approximation techniques developed for HSP and its dual Set Cover problem are closely related to the well-known boosting learning technique [16], especially in the context of the minimal committee problem looking for minimum VC-dimension correct majoritary classifier ensemble (see, e.g., [8–10]).

In addition, new efficient optimal and approximation algorithms for Hitting Set and Set Cover problems have a practical importance, e.g. in design of reliable wireless networks [15].

The Hitting Set Problem for Axis-Parallel Rectangles (HSP-APR) is a well-studied geometric setting of the HSP. This setting is also NP-hard [5] and remains intractable even for unit squares. In papers [2, 7], first polynomial time approximation schemes (PTAS) are proposed for axis-parallel squares. Paper [3] introduces 6-approximation polynomial time algorithm for the case of rectangles intersecting some axis-monotone curve. In [4], this particular case of HSP-APR is proved to be NP-hard even for the case when of rectangles intersecting a straight line and the first 4-approximation algorithm is constructed.

In this paper, we improve one of the recent results describing a polynomial time solvable subclass of this problem. Recently, Mudgal and Pandit [13] introduced an optimal polynomial time algorithm for the Hitting Set Problem for Axis Parallel Unit Squares Intersecting a given Straight Line (HSP-APUS-ISL). The theoretical importance of this result can hardly be overestimated, since almost all known geometric settings of the HSP, including extremely specific ones, are intractable. Unfortunately, this algorithm is impractical due to its incredibly high time consumption of $O(n^{145})$. In Sect. 2, we propose the improved version of the algorithm, whose complexity bound $O(n^{37})$ is still high but by more than 100 orders of magnitude less. Further, in Sect. 3, we extend this algorithm on a case of squares of different sizes (HSP-APS-ISL) and show that this problem can be solved to optimal in polynomial time for any fixed range of square sizes. The preliminary version of these results were published in [11].

Main contribution of this paper is twofold. First, we present new numerically improved version of our algorithm augmented with low-degree polynomial time preprocessing of the instance. Source code of its implementation can be downloaded from [github¹](https://github.com/EnsignDaniels/Python). Second, in Sect. 4, we provide results of numerical evaluation of the algorithm proposed. Using the classic approach to statistical model selection, we show that, for unit squares, average time complexity of the algorithm is roughly $O(n^{28})$.

2 Problem Statement

We consider the following geometric setting of the well-known Hitting Set Problem, which is called the Hitting Set Problem for Axis-Parallel Squares

¹ <https://github.com/EnsignDaniels/Python>.

Intersecting a Straight Line (HSP-APS-ISL). An instance of HSP-APS-ISL is given by a finite collection $S = \{Q_1, \dots, Q_n\}$ of axis-parallel (closed) squares in the Euclidean plane intersecting some straight line d . The goal is, for the collection S , to find a hitting set P^* of the minimum size, i.e. $|P^*| = \min\{|P| : P \subset \mathbb{R}^2, P \cap Q_j \neq \emptyset, j = 1, \dots, n\}$.

Without loss of generality we assume that the line d is defined by the equation $kx + y = 0$ for some $k \geq 1$. The collection S partitions the plane onto mutually interior-disjoint regions $\theta_1, \dots, \theta_K$ such that, any points p_1 and p_2 belong to the same region θ if and only if

$$(\forall Q_j \in S) ((p_1 \in Q_j) \iff (p_2 \in Q_j)).$$

Since each minimal hitting set contains at most one point p_k taken from any region θ_k , the initial continuous problem is polynomially equivalent to the corresponding combinatorial one, which is of finding a minimal hitting set among subsets of the finite set $\mathcal{P} = \{p_k, \dots, p_K\}, p_k \in \theta_k \setminus \bigcup_{l \neq k} \theta_l$.

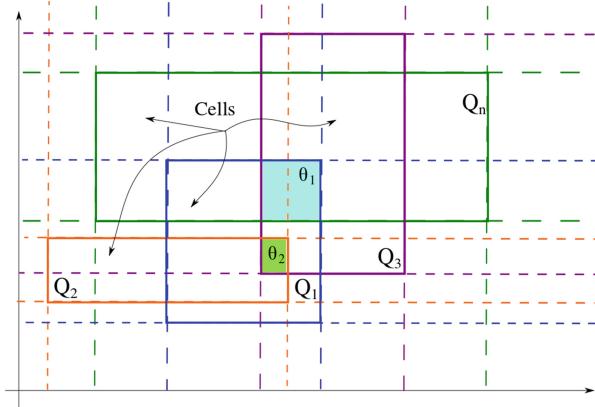


Fig. 1. The number K of different regions θ_i does not exceed the number of rectangular cells induced by the borders of Q_1, \dots, Q_n

For any collection of n axis-parallel squares (and even rectangles), the corresponding set \mathcal{P} contains at most $O(n^2)$ elements (see Fig. 1) and can be constructed efficiently. Moreover, without loss of generality, we can assume that any time the set \mathcal{P} satisfies the following assertion

Assertion 1. For any $p \in \mathcal{P}$ there is no $q \in \mathcal{P}, q \neq p$, for which $S(p) \subset S(q)$, where $S(p) = \{Q \in S : p \in Q\}$.

Indeed, points violating this condition can be filtered out in time $O(n^4)$, which is negligible small with respect to time complexities of the proposed algorithms.

3 Improved Algorithm for Unit Squares

In this section we describe a new parameterized optimal algorithm for HSP-APS-ISL and discuss its application to solving of the special case of HSP-APUS-ISL, where collection S consists of equal squares (without loss of generality, which are assumed to be unit).

We start with the similar (but not the same) notation to introduced in [13]. First, we partition the plane by straight lines l_0, \dots, l_{r+2} orthogonal to d with distance of $\sqrt{2}/2$ between each neighboring lines such that, for each square $Q_j \in S$, its center C_j is located between l_1 and l_{r+1} (hereinafter all tights are broken arbitrarily). For any $i = 0, \dots, r+1$, we denote by R_i the stripe located between l_i and l_{i+1} . Next, we introduce the notation $S_i = \{Q_j : Q_j \cap R_i \neq \emptyset\}$, $S_i^{in} = \{Q_j \in S_i : C_j \in R_i\}$, and $S_i^{out} = S_i \setminus S_i^{in}$. By construction, $S_i^{out} \subset S_{i-1}^{in} \cup S_{i+1}^{in}$.

As in [13], we assume that any stripe R_i is intersected at least by a single square Q_j . Further, we find an optimal hitting set recursively, by the dynamic programming procedure presented in Algorithm 1.

Indeed, for any $i \in 1, \dots, r$, denote $\mathcal{P}_i = \mathcal{P} \cap R_i$. Let, for $U \subset \mathcal{P}_{i-1}$ and $V \subset \mathcal{P}_i$, $T(i, U, V)$ be the size of a smallest hitting set P for $\bigcup_{l \geq i} S_l$ such that $P \cap \mathcal{P}_{i-1} = U$ and $P \cap \mathcal{P}_i = V$. Similarly to [13], we express $T(i, U, V)$ in terms of $T(i+1, U', V')$ but for a substantially smaller subsets U' and V' .

Algorithm 1. Parameterized exact DP based algorithm

Input: a collection $S = \{Q_1, \dots, Q_n\}$ of axis-parallel squares intersecting a given straight line d

Outer parameter: an upper bound q of the size of subsets to search for

Output: the minimum size hitting set P for S .

- 1: Construct a set \mathcal{P} induced by the collection S ; let $\mathcal{P}_i = \mathcal{P} \cap R_i$;
- 2: **for all** $U \subset \mathcal{P}_{r-1}$ and $V \subset \mathcal{P}_r$, s.t. $|U|, |V| \leq q$ **do**
- 3: define $\mathcal{W}_r = \{W \subset \mathcal{P}_{r+1} : |W| \leq q, U \cup V \cup W \cap Q_j \neq \emptyset (Q_j \in S_r^{in})\}$ and

$$T(r, U, V) = \begin{cases} \min\{|U \cup V \cup W| : W \in \mathcal{W}_r\}, & \text{if } \mathcal{W}_r \neq \emptyset, \\ +\infty, & \text{otherwise} \end{cases}$$

- 4: **end for**
- 5: **for all** $1 \leq i \leq r-1$ **do**
- 6: **for all** $U \subset \mathcal{P}_{i-1}$ and $V \subset \mathcal{P}_i$, s.t. $|U|, |V| \leq q$ **do**
- 7: define $\mathcal{W}_i = \{W \subset \mathcal{P}_{i+1} : |W| \leq q, U \cup V \cup W \cap Q_j \neq \emptyset (Q_j \in S_i^{in})\}$ and

$$T(i, U, V) = \begin{cases} |U| + \min\{T(i+1, V, W) : W \in \mathcal{W}_i\}, & \text{if } \mathcal{W}_i \neq \emptyset, \\ +\infty, & \text{otherwise} \end{cases}$$

- 8: **end for**
- 9: **end for**
- 10: Output

$$P = \arg \min\{T(1, U, V) : U \subset \mathcal{P}_0, V \subset \mathcal{P}_1, |U|, |V| \leq q\}.$$

Algorithm 1 has an outer parameter q , which meaning is twofold. On the first hand, q depends on size-length of the squares to hit and provides a uniform upper bound for the smallest size of a hitting set for an arbitrary S_i . On the other hand, q bounds the number of subset enumerated at each iteration of Algorithm 1. Therefore, its complexity bound can be defined in terms of q again.

The following Theorem summarizes the properties of Algorithm 1.

Theorem 1. *For $q = 6$, time complexity of Algorithm 1 is $O(n^{37})$.*

Proof. We start with the following simple fact. By construction, for any $i \in \{1, \dots, r\}$ and any $j \in S_i^{\text{in}}$, $Q_j \cap \{A, B\} \neq \emptyset$. As a consequence, for any optimal hitting set P and any $i \in \{1, \dots, r\}$, $|P_i| \leq 6$, where $P_i = P \cap R_i$. Indeed, assume by contradiction that, for some i , $|P_i| > 6$. Since $S_i \subset S_{i-1}^{\text{in}} \cup S_i^{\text{in}} \cup S_{i+1}^{\text{in}}$ and $P_i \cap Q_j = \emptyset$ for any $Q_j \notin S_i$, we can substitute P_i by an appropriate 6-point subset P'_i such that $P \cup P'_i \setminus P_i$ remains a hitting set for S and $|P'| < |P|$. The contradiction obtained with optimality of P finalizes our argument. Hence, Algorithm 1 realizing classic dynamic programming technique finds an optimal hitting set for the given collection S .

To obtain an upper bound for its running time, notice that the loop 5–9 having $r - 1 = O(n)$ iterations is the most time consuming part of Algorithm 1. In each iteration, $O(|\mathcal{P}_{i-1}|^6) \times O(|\mathcal{P}_i|^6) = O(n^{24})$ subproblems each having time complexity of $O(n^{12})$ should be solved. Therefore, the overall running time is $O(n^{37})$. \square

4 General Case of HSP-APS-ISL

By scaling, we can easily show that the result of Sect. 2 remains valid in the case of equal squares of any side-length. In this section, we extend this result to the more general case. Let a and b be the minimum and the maximum values of side-lengths of the given squares. For the same reason, assume that $a = 1$.

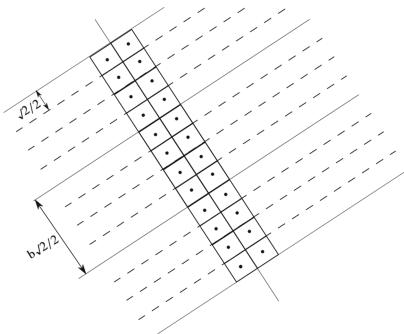


Fig. 2. Partition of the plane for $b = 4$

4.1 Case of $k = 1$

We proceed with the following observation. For $k = 1$, as in Sect. 2, any square Q of size at least 1, whose center belongs to some stripe R' of width $\sqrt{2}/2$ orthogonal to the line d , is hit by the points A and B introduced in the proof of Theorem 1. Therefore, in this case, we can adapt Algorithm 1 to take into account the squares, whose side-lengths are greater than 1.

Indeed, as above, consider stripes R_i of width $b\sqrt{2}/2$ consisting all the squares. Then, partition each of them onto $[b]$ substripes of width $\sqrt{2}/2$ (see Fig. 2) and use all other notation introduced in Sect. 2 as is. The following assertion is valid.

Theorem 2. *Let the given collection S consists of squares with side-lengths from $[1, b]$. Algorithm 1 with $q = 6[b]$ finds an optimal hitting set for this collection in time of $O(n^{6q+1}) = O(n^{36[b]+1})$.*

The argument for Theorem 2 is similar to the proof of Theorem 1. For the sake of brevity, we skip the proof.

4.2 What if $k > 1$

In this section, we show that to find an optimal solution for HSP-APS-ISL we can use Algorithm 1 again with an adjusted value of the parameter q . As above, this value is defined by the number of points needed to hit any square intersecting the line d , whose center belong to some stripe of the width $\sqrt{2}/2$. Although, for $k > 1$, points A and B (as in Fig. 2) do not hit all such squares, we can still provide a finite point collection that does.

Without loss of generality, assume that the strip R (of width $\sqrt{2}/2$) orthogonal to the line d is located symmetrically with respect to the origin. An arbitrary square Q intersecting the line d , whose center C belongs to the stripe R is called *R-centered*.

Consider finite point sequences $\{A_t\}$ and $\{B_t\}$ defined by the following equation

$$A_t = -B_t = \left[\frac{k + 2t}{2\sqrt{2(1+k^2)}}, \frac{1 - 2tk}{2\sqrt{2(1+k^2)}} \right] \quad (t \in \{-1, \dots, p\}). \quad (1)$$

Theorem 3. *For any $k > 1$, any R -centered square Q of size belonging to the range $[1, p\sqrt{2}]$ is hit by the points $A_0, \dots, A_p, B_0, B_1, \dots, B_p$.*

Proof. 1. Consider an arbitrary R -centered square Q . Theorem 3 is evidently valid if the center C of this square belongs to one of $\sqrt{2}/2$ -squares centered at A_0 or B_0 . Consider the other option. Without loss of generality, assume that C belongs to right-upper part of the stripe R (as in Fig. 2). The square Q coincides with an intersection of four closed halfplanes bordering it from the left, top, right, and bottom sides. We denote them by H_L, H_T, H_R , and H_B , respectively.

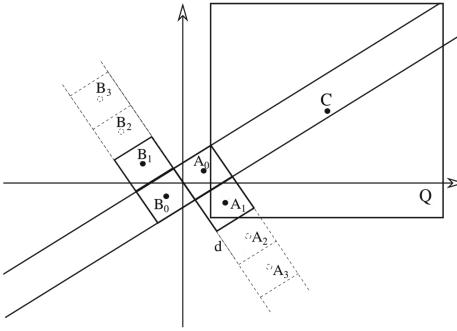


Fig. 3. Hitting of large squares by the centers of neighboring $\sqrt{2}/2$ -squares

To proceed with the argument, it is sufficient to prove that there exists a point $A_t \in Q = H_L \cap H_T \cap H_R \cap H_B$ (Fig. 3).

The inclusion $A_t \in H_T$ is valid for any $t = 0, 1, \dots, p$, since $y_{A_t} \leq y_C$ by the location assumption for the square Q . Furthermore, this assumption implies that A_{t-1} can not be located to the right of the border of H_L . Suppose, $A_{t-1} \notin H_L$ and $A_i \in H_L$ for any $i \geq t$. Now, we show that A_t is the desired point hitting the square Q . Indeed, consider the intersection point D of the line d with the vertical line visiting the point A_{t-1} . Since

$$x_D = \frac{k + 2(t-1)}{2\sqrt{2(1+k^2)}} \text{ and } kx_D + y_D = 0,$$

we obtain

$$y_{A_t} - y_D = \frac{1 - 2tk + k(k + 2(t-1))}{2\sqrt{2(1+k^2)}} = \frac{(k-1)^2}{2\sqrt{2(1+k^2)}} \geq 0.$$

Therefore, $A_t \in H_B$.

Inclusion $A_t \in H_R$ follows easily from Eq. (1). Indeed, for any $k > 1$

$$x_{A_t} - x_{A_{t-1}} = \frac{1}{2\sqrt{2(1+k^2)}} < 1/2 \leq x_C - x_{A_{t-1}},$$

since a size of the square Q is at least 1. Thus, $A_t \in H_L \cap H_T \cap H_R \cap H_B = Q$ (Fig. 4).

2. To obtain the upper side-length bound of the fittable squares, it is sufficient to calculate the minimum side-length of the R -centered square touching the point A_p by its left side (Fig. 6).

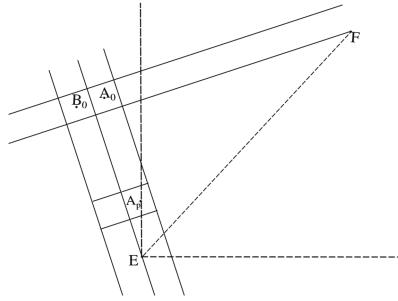


Fig. 4. Estimation of $s(\bar{k})$.

It is easy to show that this length coincides with $s = 2(x_F - x_{A_p})$, where X_F can be found from the following system

$$\begin{cases} x_E = x_{A_p} = \frac{k+2p}{2\sqrt{2(1+k^2)}} \\ kx_E + y_E = 0 \\ -x_E + y_E = z \\ -x_F + ky_F = -\frac{\sqrt{1+k^2}}{2\sqrt{2}} \\ -x_F + y_F = z, \end{cases} \quad \text{i.e. } x_F = \frac{k^3 + 2pk^2 + 2pk - 1}{2\sqrt{2}(k-1)\sqrt{1+k^2}}$$

and

$$s = \frac{k^3 + 2pk^2 + 2pk - 1}{(k-1)\sqrt{2(1+k^2)}} - \frac{k+2p}{\sqrt{2(1+k^2)}} = \frac{\sqrt{2(1+k^2)}}{2} + \frac{p\sqrt{2(1+k^2)}}{k-1}.$$

To complete our proof, we should minimize $s = s(k)$ for $k > 1$.

The derivative

$$s'(k) = \frac{\sqrt{2}}{2} \frac{k(k-1)^2 - 2p(k+1)}{(k-1)^2\sqrt{1+k^2}}$$

is vanishing if and only if

$$k^3 - 2k^2 + k = 2p(k+1). \quad (2)$$

For $p = 0$, the function $s(k)$ has no minimizers in $(1, \infty)$. The right limit

$$\lim_{k \rightarrow +1} s(k) = \inf\{s(k): k > 1\} = 1,$$

although $s(1) = +\infty$, as it follows from Subsect. 4.1.

Given by $p \geq 1$, it is sufficient to consider a few cases. If $p = 1$ we have a single root (in the feasible domain $\{k: k > 1\}$) and it is easy to see that this root is a minimizer of $s(k)$, since $s'(k)$ changes its sign at this point. Further, it can be verified that, for any $p > 1$, we also have the unique extremal point.

Denote by $\bar{k} = \bar{k}(p)$ this extremum for the given p . Using Eq. (2), we obtain

$$s(\bar{k}) = \frac{\sqrt{2}(1 + \bar{k}^2)^{3/2}}{2(1 + \bar{k})}.$$

Therefore, since $\bar{k} > 1$,

$$\frac{s(\bar{k}(p))}{p} = \frac{\sqrt{2}(1 + \bar{k}^2)^{3/2}}{\bar{k}(1 - \bar{k})^2} \geq \frac{\sqrt{2}(3/2 + \bar{k}^2)}{(\bar{k} - 1)^2} > \sqrt{2}.$$

Theorem is proved. \square

Remark 1. It is easy to verify that $\bar{k} = \bar{k}(p)$ is a monotonically increasing function and tends to $+\infty$ as $p \rightarrow +\infty$. Therefore,

$$\lim_{p \rightarrow +\infty} \frac{s(\bar{k}(p))}{p} = \lim_{\bar{k} \rightarrow +\infty} \frac{\sqrt{2}(1 + \bar{k}^2)^{3/2}}{2(1 + \bar{k})} = \sqrt{2}.$$

Applying the approach proposed in Subsect. 4.1, we obtain our final result. Indeed, let we should find the minimum hitting set for n squares intersecting the line d ; sizes of the squares belong to $[a', b']$. First, by scaling, transform their sizes to the range $[1, b]$, where $b = b'/a'$.

Further, partition the plane onto d -orthogonal stripes of width $b\sqrt{2}/2$; we call these stripes *wide*. Finally, we partition each wide stripe onto $\lceil b \rceil \sqrt{2}/2$ -width *narrow* substripes.

By construction, any square intersecting a wide stripe is centered at this or two neighboring wide stripes. Therefore, by Theorem 3, it can be hit by $q = 6\lceil b \rceil + 2\lceil b/\sqrt{2} \rceil$, and the optimal hitting set can be found by Algorithm 1 using this value of q . Hence, we proved the following theorem.

Theorem 4. For any constant c and any square collection with size-range $[a, c\bar{a}]$, the problem HSP-ASP-ISL can be solved to optimality in time $O(n^{6q+1})$, where $q = 6\lceil c \rceil + 2\lceil c/\sqrt{2} \rceil$.

Remark 2. Results of Theorems 2 and 4 shows that HSP-APS-ISL is polynomial solvable for any fixed range of squares, since the running time bound of Algorithm 1 in this case is $O(n^{6(6\lceil c \rceil + 2\lceil c/\sqrt{2} \rceil) + 1})$. Unfortunately, the question of constructing for this problem an FPT algorithm having parameterized complexity bound like $f(c) \cdot n^{O(1)}$ still remains open.

5 Numerical Evaluation

The numerical experiment described in this section is motivated by the following observation. Any time, when the raw point set \mathcal{P} (assigned to the given instance of HSPAPSISL) was distilled to filter out points violating Assertion 1, its size was decreased significantly (Fig. 5). But, having no theoretical bounds for such a decreasing, in Theorems 1–3, we use a rough upper bound $|\mathcal{P}| = O(n^2)$ during time complexity estimation of Algorithm 1. Therefore, we decided to evaluate numerically the dependency between an average value of $|\mathcal{P}|$ and instance size on a random sample of instances of the problem in the simplest case of unit squares.

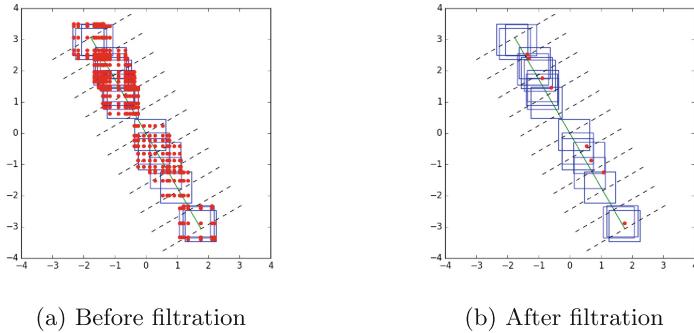


Fig. 5. Set of axis-parallel squares S and the set \mathcal{P}

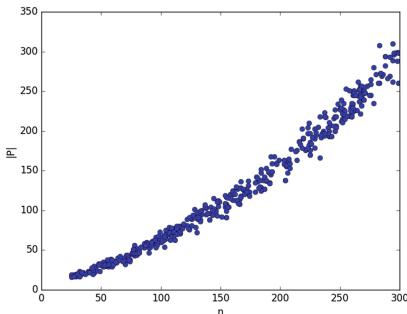


Fig. 6. Scatter-plot for n and $|\mathcal{P}|$

Experiment setup. We take the fixed value of $k = \sqrt{3}$, sample at random 500 instance sizes (n_1, \dots, n_{500}) , $n_i \in \{25, \dots, 300\}$ and consider appropriate instances of HSP-APS-ISL, each of them is defined by n_i unit squares, whose centers are taken from the uniform distribution at the stripe of fixed length $N = 50$ centered at the line d . For any instance, we produce a filtered set \mathcal{P} and measure its size. Such a way, we obtain the scatter-plot as follows (Fig. 6).

To obtain the best fit dependence, we evaluate statistically several models and take the most confident in terms of R^2 criterion (Fig. 7). Further, residual analysis helps us to take the most adequate model $|\mathcal{P}| \approx 0.056n^{1.5}$, where the coefficient is estimated with the confidence 95%. Taking into account the dependency evaluated, we obtain $O(n^{28})$ as an average time complexity of Algorithm 1 for the case of unit squares.

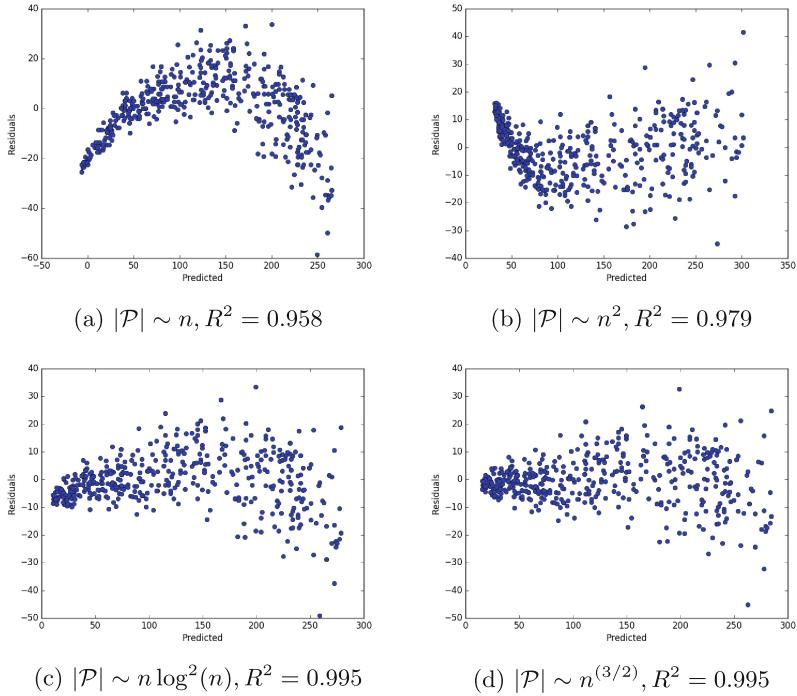


Fig. 7. Evaluation for different dependencies between $|\mathcal{P}|$ and n

6 Conclusion

Improved version of the optimal polynomial time hitting set construction algorithm for axis-parallel squares intersecting the given straight line introduced in [13] is proposed. Our modification has a significantly better upper time complexity bound by 100 orders of magnitude.

Also, we have managed to extend this algorithm to the case of non-unit squares and show that the problem can be solved to optimality in polynomial time for any fixed range of squares.

However, it would be interesting to establish the complexity status of the considered problem in the case, where this parameter is a part of an instance. Also, it can be interesting to apply the recent results on shallow cell complexity (see, e.g. [14]) to refine complexity bounds of the algorithms proposed.

Acknowledgements. This research was supported by Russian Foundation for Basic Research, grants no. 16-07-00266 and 17-08-01385, and Complex Program of Ural Branch of RAS, grant no. 15-7-1-23.

References

1. Brönnimann, H., Goodrich, M.T.: Almost optimal set covers in finite VC-dimension. *Discrete Comput. Geom.* **14**(4), 463–479 (1995)
2. Chan, T.M.: Polynomial-time approximation schemes for packing and piercing fat objects. *J. Algorithms* **46**(2), 178–189 (2003)
3. Chepoi, V., Felsner, S.: Approximating hitting sets of axis-parallel rectangles intersecting a monotone curve. *Comput. Geom.* **46**(9), 1036–1041 (2013)
4. Correa, J., Feuilloley, L., Pérez-Lantero, P., Soto, J.A.: Independent and hitting sets of rectangles intersecting a diagonal line: algorithms and complexity. *Discrete Comput. Geom.* **53**(2), 344–365 (2015)
5. Fowler, R.J., Paterson, M.S., Tanimoto, S.L.: Optimal packing and covering in the plane are NP-complete. *Inf. Process. Lett.* **12**(3), 133–137 (1981)
6. Haussler, D., Welzl, E.: Epsilon-nets and simplex range queries. *Discrete Comput. Geom.* **2**(2), 127–151 (1987)
7. Hochbaum, D., Maass, W.: Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM* **32**(1), 130–136 (1985)
8. Khachay, M.: Committee polyhedral separability: complexity and polynomial approximation. *Mach. Learn.* **101**(1), 231–251 (2015)
9. Khachay, M., Poberii, M.: Complexity and approximability of committee polyhedral separability of sets in general position. *Informatica* **20**(2), 217–234 (2009)
10. Khachay, M., Pobery, M., Khachay, D.: Integer partition problem: theoretical approach to improving accuracy of classifier ensembles. *Int. J. Artif. Intell.* **13**(1), 135–146 (2015)
11. Khachay, M., Khachay, D.: On parameterized complexity of the Hitting Set Problem for axis-parallel squares intersecting a straight line. *Ural Math. J.* **2**, 117–126 (2016)
12. Matoušek, J.: Lectures on Discrete Geometry. Springer, New York (2002). <https://doi.org/10.1007/978-1-4613-0039-7>
13. Mudgal, A., Pandit, S.: Covering, hitting, piercing and packing rectangles intersecting an inclined line. In: Lu, Z., Kim, D., Wu, W., Li, W., Du, D.-Z. (eds.) COCOA 2015. LNCS, vol. 9486, pp. 126–137. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26626-8_10
14. Mustafa, N.H., Varadarajan, K.: Epsilon-approximations and epsilon-nets. CoRR, abs/1702.03676 (2017)
15. Ramakrishnan, S., El Emam, I.M.M.: Wireless Sensor Networks: From Theory to Applications. Taylor & Francis, CRC Press, New York (2014)
16. Schapire, R., Freund, Y.: Boosting: Foundations and Algorithms. MIT Press, Cambridge (2012)
17. Vapnik, V., Chervonenkis, A.: On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl.* **16**, 264–280 (1971)

Polynomial Time Solvable Subclass of the Generalized Traveling Salesman Problem on Grid Clusters

Michael Khachay¹✉ and Katherine Neznakhina²

¹ Krasovsky Institute of Mathematics and Mechanics, Ural Federal University,
Ekaterinburg, Russia
mkhachay@imm.uran.ru

² Omsk State Technical University, Omsk, Russia
eneznakhina@yandex.ru

Abstract. The Generalized Traveling Salesman Problem on Grid Clusters (GTSP-GC) is the geometric setting of the famous Generalized Traveling Salesman Problem, where the nodes of a given graph are points on the Euclidean plane and the clusters are defined implicitly by the cells of a unit grid. The problem in question is strongly NP-hard but can be approximated in polynomial time with a fixed ratio. In this paper we describe a new non-trivial polynomially solvable subclass of GTSP-GC. Providing new min-max guarantee for the optimal clustering loss in one-dimensional 2-medians problem, we show that any instance of this subclass has a quasi-pyramidal optimal route, which can be found by dynamic programming in polynomial time.

Keywords: Generalized traveling salesman · Grid clusters
Pyramidal tour

1 Introduction

The Traveling Salesman Problem (TSP) is the famous combinatorial optimization problem having many valuable applications in operations research and attracting interest of scientists for decades (see, e.g. [11, 18]).

It is known that TSP is strongly NP-hard and hardly approximable in its general setting [19]. Although, the problem remains intractable in metric and Euclidean settings, it can be approximated well in these cases admitting fixed-ratio algorithms for an arbitrary metric [7] and Polynomial Time Approximation Schemes (PTAS) for Euclidean spaces of any fixed dimension [1]. Many generalizations of TSP, e.g. Cycle Cover Problem [10, 12, 13, 16], Peripatetic Salesman Problem [2, 9], have the similar approximation behaviour.

Algorithmic issues of finding optimal restricted tours, for several kinds of restrictions, e.g. precedence constraints, are also actively investigated (see., e.g. [3, 6]). Among others, restriction of TSP to considering so called *pyramidal tours* (see, e.g. [5]) seems to be especially popular. Pyramidal tour respects

the initial order defined on the nodeset of a given graph and has the form $1 = v_{i_1}, v_{i_2}, \dots, v_{i_r} = n, v_{i_{r+1}}, \dots, v_{i_n}$ such that $v_{i_j} < v_{i_{j+1}}$ for any $j \in \{1, \dots, r-1\}$ and $v_{i_j} > v_{i_{j+1}}$ for any $j \in \{r+1, \dots, n-1\}$. It is widely known [15] that an optimal pyramidal tour can be found in time of $O(n^2)$ for any weighting function. In papers [8, 17], several generalizations of pyramidal tours, for which optimal tour can be found efficiently were introduced. Recently it was shown [4] that, for the Euclidean setting, optimal pyramidal tour can be found in time $O(n \log^2 n)$. Despite their fame, pyramidal tours have one shortcoming. Known settings of TSP and its generalizations, for which existence of optimal pyramidal tours is proven, are very rare. Actually, they are exhausted with settings satisfying the well known sufficient conditions by Demidenko and van der Veen (see, e.g. [11]).

Contribution of this paper is two-fold. At first, we introduce (in Sect. 2) the notion of l -quasi-pyramidal tour extending the classic notion of pyramidal tour to the case of Generalized Traveling Salesman Problem (GTSP) and show that optimal l -quasi-pyramidal tour can be found in time $O(n^3)$ for any fixed l . At second, we describe (in Sect. 3) a non-trivial polynomially solvable subclass of GTSP, for which the existence of optimal l -quasi-pyramidal tour (for some fixed l) is proved.

2 Quasi-Pyramidal Tours

We proceed with the common setting of the Generalized Traveling Salesman Problem (GTSP). Instance of the GTSP is defined by complete edge-weighted graph $G = (V, E, w)$ with weighting function $w: E \rightarrow \mathbb{R}_+$, and by a given partition $V_1 \cup \dots \cup V_k = V$ of the nodeset $V = V(G)$ of graph G . Feasible solutions are cyclic tours v_{i_1}, \dots, v_{i_k} visiting each cluster V_i once. Hereinafter, we call such routes *Clustered Hamiltonian tours* or *CH-tours*. The problem is to find a CH-tour of the minimum weight¹.

In this section, we extend the well-known notion of a pyramidal tour to the case of partial orders defined implicitly by the orderings of clusters. Indeed, linear ordered finite set (V_1, \dots, V_k) of clusters induces a partial order on the nodeset V of the graph G as follows, for any $u \in V_i$ and $v \in V_j$, $u \prec v$ if $i < j$.

Definition 1. Let τ be a CH-tour $v_1, v_{i_1}, \dots, v_{i_r}, v_k, v_{j_{k-r-2}}, \dots, v_{j_1}$ such that $v_t \in V_t$ for any t . We call τ an l -quasi-pyramidal tour, if $i_p - i_q \leq l$ and $j_{p'} - j_{q'} \leq l$ for any $1 \leq p < q \leq r$ and $1 \leq p' < q' \leq k-r-2$.

The following theorem extends the results proposed in [17, Theorem 3.6] for the classical TSP.

Theorem 1. For any weighting function $w: E \rightarrow \mathbb{R}_+$, a minimum cost l -quasi-pyramidal CH-tour can be found in time of $O(4^l n^3)$.

¹ To the sake of brevity, we restrict ourselves to the case of undirected graphs. Our argument can be easily extended to the case of digraphs and asymmetric weighting functions w .

We present a short sketch of the proof of Theorem 1 postponing its full version to the forthcoming paper. First of all, we introduce some necessary notation.

For any integers $i > j$, we use common shortcuts $[j, i]$, $[j, i)$, and (j, i) for intersections with \mathbb{N} of the sets $\{j, \dots, i\}$, $\{j, \dots, i - 1\}$, $\{j + 1, \dots, i - 1\}$, respectively. For any nodes $u \in V_i$ and $v \in V_j$, $i \neq j$, and an arbitrary subset $S \subset [i - l, i) \setminus \{1, j\}$ or $S \subset [j - l, j) \setminus \{1, i\}$, let $g(v, S, u)$ be the weight of a shortest $(|S| + 1)$ -edge path from u to v visiting all the clusters $\{V_t : t \in S\}$ (see Fig. 1). Values of the function g can be easily calculated recursively, since $g(v, \emptyset, u) = w(\{v, u\})$ and

$$g(v, S, u) = \begin{cases} \min_{m \in S} \min_{v' \in V_m} \{g(v, S \setminus \{m\}, v') + w(\{v', u\})\}, & \text{if } S \subseteq [j - l, j) \setminus \{1, i\}, \\ \min_{m \in S} \min_{v' \in V_m} \{w(\{v, v'\}) + g(v', S \setminus \{m\}, u)\}, & \text{if } S \subseteq [i - l, i) \setminus \{1, j\}. \end{cases} \quad (1)$$

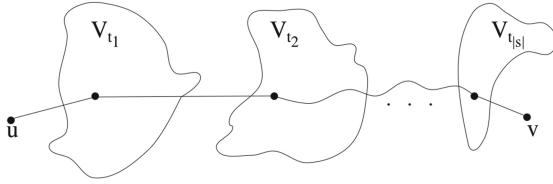


Fig. 1. Path from u to v through the clusters V_{t_j} , $t_j \in S$.

Further, for any $1 \leq j < i \leq k$, let $f(u, v, T)$ be the weight of a shortest path P from $u \in V_i$ to $v \in V_j$ visiting all the clusters with numbers from $[1, i) \cup [1, j) \setminus T$, where $T \subseteq [i - l, i) \cup [j - l, j) \setminus \{1, i, j\}$, and the path P has the form

$$u = v_{i_0}, v_{i_1}, \dots, v_{i_r} = \bar{v} = v_{j_0}, v_{j_1}, \dots, v_{j_s} = v,$$

for pairwise different indexes $i_0, \dots, i_r, j_1, \dots, j_s$, such that $\bar{v} \in V_1$, $i_t < i$ for $1 \leq t \leq r$, $j_{t'} < j$ for $0 \leq t' \leq s - 1$, and

$$\begin{aligned} i_q - i_p &\leq l, \quad (0 < p < q \leq r), \\ j_{p'} - j_{q'} &\leq l, \quad (0 \leq p' < q' \leq s). \end{aligned}$$

As the function g , values of the function f can be obtained recursively. We start with values $f(u, v, (1, t)) = w(\{u, v\})$ for any $u \in V_1$ and $v \in V_t$, $2 \leq t \leq l + 2$. All other necessary values $f(u, v, T)$ for any $u \in V_i$, $v \in V_j$ and any $T \subset [i - l, i) \cup [j - l, j) \setminus \{1, i, j\}$ can be computed in ascending order by i and $j < i$ as follows. Let m be the maximum number of cluster (excluding i and j) visited by the path P . If $m > j$, then $f(u, v, T)$ can be calculated by formula

$$f(u, v, T) = \min_{S \subseteq [m-l, m) \setminus (T \cup \{1, j\})} \min_{u' \in V_m} \{g(u, S, u') + f(u', v, T \cup S)\}, \quad (2)$$

otherwise

$$f(u, v, T) = \min \begin{cases} \min_{S \subseteq [m-l, m] \setminus (T \cup \{1\})} \min_{u' \in V_m} \{g(u, S, u') + f(u', v, T \cup S)\}, & \text{if } m \in \{i_1, \dots, i_{r-1}\} \\ \min_{S \subseteq [m-l, m] \setminus (T \cup \{1\})} \min_{u' \in V_m} \{f(u, u', T \cup S) + g(u', S, v)\} & \text{otherwise.} \end{cases} \quad (3)$$

Finally, we obtain $f(u, v, T)$ for any $u \in V_k$ and $v \in V_{k-1}$ and for any $T \subseteq [k-l-1, k-1]$. The weight of optimal l -quasi-pyramidal tour (see, Fig. 2) is given by

$$\min_{T \subseteq [k-l-1, k-1]} \min_{u \in V_k} \min_{v \in V_{k-1}} \{f(u, v, T) + g(v, T, u)\}. \quad (4)$$

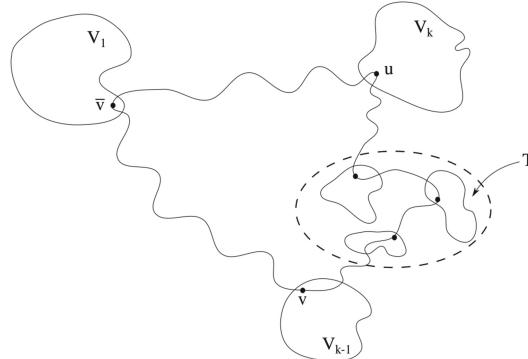


Fig. 2. Constructing a minimum weight l -quasi-pyramidal tour

Compute a naïve upper bound for time complexity of the algorithm. At first, we calculate the necessary values $g(v, S, u)$ by formula (1) in time $O(2^l n^3)$. Then, the initial values $f(u, v, (1, t))$ can be computed in $O(n^2)$. Further, for any fixed u, v and T , the complexity of Eqs. (2) and (3) do not exceed $O(2^l n)$. Since, formulas (2) and (3) are invoked at most $O(2^l n^2)$ times, the overall time complexity bound is $O(4^l n^3)$, which completes the sketch of our proof.

Remark 1. Since any CH-tour τ is l -quasi-pyramidal for some $l = l(\tau) \in [0, n]$, the result of Theorem 1 can be considered in the context of the parameterized complexity. Actually, Theorem 1 claims that, in the most general setting, GTSP is fixed-parameter tractable with respect to parametrization induced by quasi-pyramidal tours.

3 Polynomial Time Solvable Subclass of GTSP on Grid Clusters

In this section, we describe polynomially solvable subclass of Generalized Traveling Salesman Problem on Grid Clusters, GTSP-GC for short. In this special

case of the GTSP, an undirected edge-weighted graph $G = (V, E, w)$ is given where the set of vertices V correspond to a set of points in the planar rectangular grid. Every nonempty 1×1 cell of the grid forms a cluster. The weighting function is induced by distances between the appropriate points with respect to some metric. To put it simple, we consider Euclidean distances, but the similar results can be easily obtained for some other metrics, e.g. for l_1 . In Fig. 3, we present an instance of the Euclidean GTSP-GC with 6 clusters.

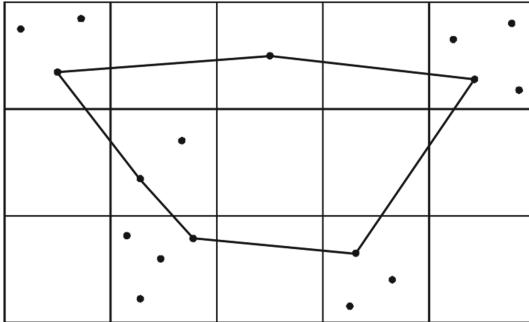


Fig. 3. An instance of the Euclidean GTSP-GC and its optimal solution

It is known that for two special cases of the problem, when the number k of clusters is $O(\log n)$ or $n - O(\log n)$, polynomial time approximation schemes (PTAS) were proposed [14]. Meanwhile, the question of systematic description of polynomial time solvable subclasses of GTSP-GC, which is closely related to complexity analysis of Hamiltonian cycle problem on grid graphs, is still far from its complete answer.

Let H and W be *height* and *width* (number of rows and columns) of the given grid, respectively. We consider a special case of the GTSP-GC, for which one of these parameters, say H does not exceed 2 (while another one is unbounded). We call this case GTSP-GC(H2). We show that any instance of GTSP-GC(H2) has an optimal l -quasi-pyramidal CH-tour for some l independent on n . Therefore, this subclass of GTSP-GC is polynomially solvable due to Theorem 1.

Our argument is based on the introduced *tour straightening transformation*, which is closely related to the well-known class of *local search heuristics*. To describe the transformation, assign to columns of the grid defining the given instance of GTSP-GC(H2), integer numbers $1, 2, \dots, W$ (from the left to the right). Consider an arbitrary CH-tour τ . Assigning to each node v_i of τ the number c_i of the column it belongs, obtain a sequence σ of column numbers presented in the order induced by the tour τ . Without loss of generality, assume that σ has the form

$$1 = c_1, c_2, \dots, c_r = W, c_{r+1}, \dots, c_s = 1 \quad (5)$$

for some appropriate numbers r and s .

Suppose, for some integer number t , whose value will be specified later, there exist indexes

$$1 \leq p < q \leq r, \text{ such that } c_p - c_q \geq t - 1, \text{ or} \quad (6)$$

$$r + 1 \leq p' < q' \leq s, \text{ such that } c_{q'} - c_{p'} \geq t - 1. \quad (7)$$

In this case, we say that the tour τ has t -zigzag. Obviously, any l -quasi-pyramidal tour contains no t -zigzags, for any $t \geq l$. Algorithm 1 replaces all segments of the tour τ having t -zigzags with subtours of the special kind (see. Fig. 5).

Algorithm 1. Tour straightening transformation

Outer Parameter: t .

Input: an instance of GTSP-GC(H2) and a CH-tour τ .

Output: a CH-tour τ' without t -zigzags.

- 1: set $\tau' := \tau$
 - 2: **while** τ' has t -zigzag **do**
 - 3: assume that equation (6) is valid (without loss of generality, we assume that $c_p - c_q = t - 1$), the case of (7) can be treated similarly;
 - 4: let C be the set of columns with numbers c_q, \dots, c_p (see Fig. 4);
 - 5: let $Y = (y_1, \dots, y_{2t+4})$ be the ordinate sequence of nodes visited by τ in C augmented by ordinates of left and right crossing points;
 - 6: find an optimal 2-medians clustering for Y with medians m_1 and m_2 ;
 - 7: replace segments of the tour τ' belonging to C by horizontal lines at height m_1 and m_2 connected to all points mentioned in Step 5 by line segments (Fig. 5)
 - 8: **end while**
 - 9: output the CH-tour τ' .
-

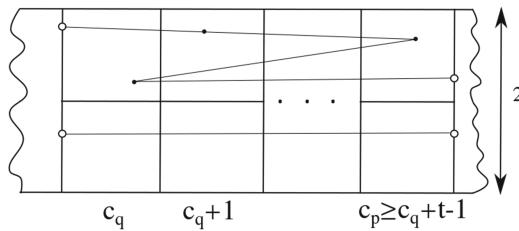


Fig. 4. Segment of τ with t -zigzag

To specify the value of t , notice that the weight of eliminated segments of τ has an evident lower bound $t + 2(t - 1) + t - 2 = 4t - 4$. Meanwhile, the weight of their replacement in Step 7 at any iteration of Algorithm 1 is at most $2t + 2F(Y, [0, 2])$, where $F(Y, S)$ is an optimum value of 2-medians clustering objective function for a sample Y taken from a line segment S .

To estimate an upper bound for $F(Y, S)$ we need the following Lemma.

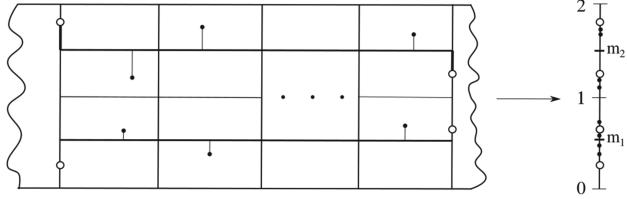


Fig. 5. Replacing t -zigzag with a special tour segments

Lemma 1. For any sample $\xi = (p_1, \dots, p_n)$, $p_i \in [0, 1]$ there exist numbers m_1 and $m_2 \in [0, 1]$ such that

$$F(\xi, [0, 1]) = \sum_{i=1}^n \min\{|p_i - m_1|, |p_i - m_2|\} \leq n/6. \quad (8)$$

Lemma 1 seems to be a folk theorem. We present a short sketch of its proof in Appendix A.

Getting back to discussion of Algorithm 1, we obtain from Lemma 1 that $F(Y, [0, 2]) \leq 2 \cdot 1/6(2t + 4)$. Therefore, at any iteration of Algorithm 1, the tour τ' becomes cheaper if $2t + 4t/3 + 8/3 \leq 4t - 4$, i.e. $t \geq 10$.

Further, let cells of the grid be ordered as in Fig. 6 (i.e., top-down and left-right). For $t = 10$, any CH-tour of the given GTSP-GC(H2) instance can be transformed to l -quasi-pyramidal CH-tour for $l = 20$ without increasing its weight. Hence, we are proved the following theorem.

Theorem 2. Any instance of GTSP-GC(H2) has an optimal 20-quasi-pyramidal CH-tour.

As a consequence of Theorems 1 and 2, we obtain that GTSP-GC(H2) can be solved to optimality in time $O(n^3)$.

1	3	5		k-1
2	4	6		k

Fig. 6. Cluster ordering

4 Conclusion

In this paper, a new notion of l -quasi-pyramidal tour extending the classic notion of pyramidal tour is introduced. We show that, similar to the case of pyramidal

tours and TSP, an optimal l -quasi-pyramidal tour for the Generalized Traveling Salesman Problem can be found efficiently (for an arbitrary weighting function). Also, we describe a non-trivial polynomially solvable geometric special case of GTSP. Each instance of the problem in question has an l -quasi-pyramidal tour as an optimal solution. Actually, an instance of this problem is defined by unit 2-row rectangular grid on the Euclidean plane. As for open questions, it would be interesting to extend the result obtained to the case of GTSP-GC(Hh) defined by a grid of an arbitrary fixed height h .

Acknowledgments. This research was supported by Russian Science Foundation, project no. 14-11-00109.

A Proof of Lemma 1

Indeed, consider the following two-gamer zero-sum antagonistic game. The first player choose an n -length sample $\xi = (p_1, \dots, p_n)$ from $[0, 1]$. The second player proposes a 2-partition $C_1 \cup C_2 = [1, n]$. Payoff function

$$F(\xi, (C_1, C_2)) = \sum_{i \in C_1} |p_i - m_1| + \sum_{i \in C_2} |p_i - m_2| = \sum_{i=1}^n \min\{|p_i - m_1|, |p_i - m_2|\},$$

where m_1 and m_2 are medians of subsamples $\xi_1 = (p_i : i \in C_1)$ and $\xi_2 = (p_i : i \in C_2)$, respectively.

It is easy to verify that this game has no value. To obtain its lower value, notice that equation

$$\sum_{i \in C} |p_i - m| = \begin{cases} \sum_{i=k+1}^{2k} p_i - \sum_{i=1}^k p_i, & \text{if } |C| = 2k, \\ \sum_{i=k+2}^{2k+1} p_i - \sum_{i=1}^k p_i, & \text{if } |C| = 2k+1 \end{cases}$$

is valid for any non-empty index set C , median m , and sample

$$p_1 \leq p_2 \leq \dots \leq p_{|C|}.$$

Therefore $\sup_{\xi} \inf_{C_1, C_2} F(\xi, (C_1, C_2))$ is an optimum of the appropriate linear program

$$\alpha = \max u$$

$$\begin{aligned} \text{s.t. } & \sum_{i=\lceil |C_1|/2 \rceil + 1}^{|C_1|} p_i - \sum_{i=1}^{\lfloor |C_1|/2 \rfloor} p_i + \sum_{i=\lceil |C_2|/2 \rceil + 1}^{|C_2|} p_{i+|C_1|} - \sum_{i=1}^{\lfloor |C_2|/2 \rfloor} p_{i+|C_1|} \geq u \\ & (C_1 \cup C_2 = [1, n]) \\ & 0 \leq p_1, \dots, p_n \leq 1 \end{aligned} \tag{9}$$

Using one of common linear programming techniques, e.g. variable elimination, it is easy to show that $\alpha = n/6$. Lemma is proved.

References

1. Arora, S.: Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. ACM* **45**, 753–782 (1998)
2. Baburin, A., Della Croce, F., Gimadi, E.K., Glazkov, Y.V., Paschos, V.T.: Approximation algorithms for the 2-peripatetic salesman problem with edge weights 1 and 2. *Discret. Appl. Math.* **157**(9), 1988–1992 (2009)
3. Balas, E.: New classes of efficiently solvable generalized traveling salesman problems. *Ann. Oper. Res.* **86**, 529–558 (1999)
4. de Berg, M., Buchin, K., Jansen, B.M.P., Woeginger, G.: Fine-grained complexity analysis of two classic TSP variants. In: Chatzigiannakis, I., Mitzenmacher, M., Rabani, Y., Sangiorgi, D. (eds.) 43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016). Leibniz International Proceedings in Informatics (LIPIcs), vol. 55, pp. 5:1–5:14. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2016). <http://drops.dagstuhl.de/opus/volltexte/2016/6277>
5. Burkard, R.E., Deineko, V.G., van Dal, R., van der Veen, J.A.A., Woeginger, G.J.: Well-solvable special cases of the traveling salesman problem: a survey. *SIAM Rev.* **40**(3), 496–546 (1998)
6. Chentsov, A.G., Khachai, M.Y., Khachai, D.M.: Proc. Steklov Inst. Math. **295**(Suppl 1), 38–46 (2016). <https://doi.org/10.1134/S0081543816090054>
7. Christofides, N.: Worst-case analysis of a new heuristic for the traveling salesman problem. In: Symposium on New Directions and Recent Results in Algorithms and Complexity, p. 441 (1975)
8. Enomoto, H., Oda, Y., Ota, K.: Pyramidal tours with step-backs and the asymmetric traveling salesman problem. *Discret. Appl. Math.* **87**(1–3), 57–65 (1998)
9. Gimadi, E.K., Glazkov, Y., Tsidulko, O.Y.: Probabilistic analysis of an algorithm for the m-planar 3-index assignment problem on single-cycle permutations. *J. Appl. Ind. Math.* **8**(2), 208–217 (2014)
10. Gimadi, E.K., Rykov, I.A.: On the asymptotic optimality of a solution of the euclidean problem of covering a graph by m nonadjacent cycles of maximum total weight. *Dokl. Math.* **93**(1), 117–120 (2016)
11. Gutin, G., Punnen, A.P.: The Traveling Salesman Problem and Its Variations. Springer, Boston (2007). <https://doi.org/10.1007/b101971>
12. Khachai, M., Neznakhina, E.: Approximability of the problem about a minimum-weight cycle cover of a graph. *Dokl. Math.* **91**(2), 240–245 (2015)
13. Khachay, M., Neznakhina, K.: Approximability of the minimum-weight k-size cycle cover problem. *J. Global Optim.* **66**(1), 65–82 (2016). <https://doi.org/10.1007/s10898-015-0391-3>
14. Khachay, M., Neznakhina, K.: Towards a PTAS for the generalized TSP in grid clusters. *AIP Conf. Proc.* **1776**(1), 050003 (2016)
15. Klyaus, P.: Generation of testproblems for the traveling salesman problem. Preprint Inst. Mat. Akad. Nauk. BSSR (16) (1976). (in Russian)
16. Neznakhina, E.: PTAS for Min- k -SCCP in Euclidean space of arbitrary fixed dimension. *Proc. Steklov Inst. Math.* **295**(1), 120–130 (2016). <https://doi.org/10.1134/S0081543816090133>
17. Oda, Y., Ota, K.: Algorithmic aspects of pyramidal tours with restricted jump-backs. *Interdisc. Inf. Sci.* **7**(1), 123–133 (2001)

18. Pardalos, P., Du, D., Graham, R.: *Handbook of Combinatorial Optimization*. Springer, New York (2013)
19. Sahni, S., Gonzales, T.: P-complete approximation problems. *J. ACM* **23**, 555–565 (1976)

Stabbing Line Segments with Disks: Complexity and Approximation Algorithms

Konstantin Kobylkin^{1,2(✉)}

¹ Institute of Mathematics and Mechanics, Ural Branch of RAS,
Sophya Kovalevskaya str. 16, 620990 Ekaterinburg, Russia

kobylkinks@gmail.com

² Ural Federal University, Mira str. 19, 620002 Ekaterinburg, Russia

Abstract. Computational complexity and approximation algorithms are reported for a problem of stabbing a set of straight line segments with the least cardinality set of disks of fixed radii $r > 0$ where the set of segments forms a straight line drawing $G = (V, E)$ of a planar graph without edge crossings. Close geometric problems arise in network security applications. We give strong NP-hardness of the problem for edge sets of Delaunay triangulations, Gabriel graphs and other subgraphs (which are often used in network design) for $r \in [d_{\min}, \eta d_{\max}]$ and some constant η where d_{\max} and d_{\min} are Euclidean lengths of the longest and shortest graph edges respectively. Fast $O(|E| \log |E|)$ -time $O(1)$ -approximation algorithm is proposed within the class of straight line drawings of planar graphs for which the inequality $r \geq \eta d_{\max}$ holds uniformly for some constant $\eta > 0$, i.e. when lengths of edges of G are uniformly bounded from above by some linear function of r .

Keywords: Computational complexity · Approximation algorithms
Hitting set · Continuous Disk Cover · Delaunay triangulations

1 Introduction

Numerous applications from security, sensor placement and robotics lead to computational geometry problems in which one needs to find the smallest cardinality set C of points on the plane having bounded (in some sense) visibility area such that each piece of the boundary of a given geometric object or any part of the complex (i.e. set of edges or faces) of a plane graph is within visibility area of some point from C , see e.g. [5, 12]. Refining complexity statuses and designing approximation algorithms for these problems is still an area of active research. In this paper complexity and approximability are studied of the following problem. INTERSECTING PLANE GRAPH WITH DISKS (IPGD): given a straight line drawing (or a plane graph) $G = (V, E)$ of an arbitrary simple¹ planar graph without

This work was supported by Russian Science Foundation, project 14-11-00109.

¹ A graph without loops and parallel edges.

edge crossings and a constant $r > 0$, find the smallest cardinality set $C \subset \mathbb{R}^2$ of points (disk centers) such that each edge $e \in E$ is within Euclidean distance r from some point $c = c(e) \in C$ or, equivalently, the disk of radius r centered at c intersects e .

The IPGD abbreviation is used throughout our paper to denote the above problem for simplicity of presentation. Applications of complexity and algorithmic analysis of the IPGD problem come from network security. More specifically, IPGD represents the following model in which we are to evaluate vulnerability of some physical network to simultaneous technical failures caused by natural (e.g. floods, fire, electromagnetic pulses) and human sources. In this model network nodes are modeled by points on the plane while its physical links are given in the form of straight line segments. A catastrophic event (threat) is usually localized in a particular geographical area and modeled by a disk of some fixed radius $r > 0$. A threat impacts a network link when the corresponding disk and segment intersect. Evaluation of the network vulnerability can be posed in the form of finding the minimum number of threats along with their positions that cause all network links to be broken. Thus, it brings us to the IPGD problem assuming that network links are geographically non-overlapping. A similar setting is considered in [1] with the fixed number of threats. Furthermore, in [12] a close geometric problem is considered called the Art Gallery problem where point coverage area is affected by boundaries of its neighbouring geometric objects whereas point has circular visibility area in the case of the IPGD problem.

In this paper computational complexity and approximability of IPGD are studied for simple plane graphs with either $r \in [d_{\min}, d_{\max}]$ or $r = \Omega(d_{\max})$ where d_{\max} and d_{\min} are Euclidean lengths of the longest and shortest edges of G . Our emphasis is on those classes of simple plane graphs that are defined by some distance function, namely, on Delaunay triangulations, some of their connected subgraphs, e.g. for Gabriel graphs. These graphs are often called *proximity* graphs. Delaunay triangulations are plane graphs which admit efficient geometric routing algorithms [4], thus, representing convenient network topologies. Gabriel graphs arise in modeling wireless networks [14].

1.1 Related Work

IPGD is related to several well-known combinatorial optimization problems. First, we have the Continuous Disk Cover (CDC) problem for the case of IPGD where G consists of isolated vertices, i.e. when segments from E are all of zero length. Strong NP-hardness is well known for CDC [9]. Second, IPGD coincides with known VERTEX COVER problem for $r = 0$. Third, it is the special case of the geometric HITTING SET problem on the plane.

HITTING SET: given a family \mathcal{N} of sets on the plane and a set $U \subseteq \mathbb{R}^2$, find the smallest cardinality set $H \subseteq U$ such that $N \cap H \neq \emptyset$ for every $N \in \mathcal{N}$.

IPGD coincides with HITTING SET if we set $\mathcal{N} := \mathcal{N}_r(E) = \{N_r(e)\}_{e \in E}$ and $U := \mathbb{R}^2$ where $N_r(e) = B_r(0) + e = \{x + y : x \in B_r(0), y \in e\}$ is Euclidean r -neighbourhood of e having form of Minkowski sum and $B_r(x)$ is the disk of radius

r centered at $x \in \mathbb{R}^2$. An *aspect ratio* of a closed convex set N with $\text{int } N \neq \emptyset$ ² coincides with the ratio of the minimum radius of the disk which contains N to the maximum radius of the disk which is contained in N . For example, each set $N_r(e)$ (also called by object in the sequel) has aspect ratio equal to $1 + \frac{d(e)}{2r}$, where $d(e)$ is Euclidean length of edge $e \in E$. APX-hardness of the discrete³ HITTING SET problem is given for families of axis-parallel rectangles, generally, with unbounded aspect ratio [6], and for families of triangles of bounded aspect ratio [13].

1.2 Results

Our results report complexity and approximation algorithms for the IPGD problem within several classes of plane graphs under different assumptions on r . Let S be a set of n points in general position on the plane no four of which are cocircular. We call a plane graph $G = (S, E)$ a *Delaunay triangulation* if $[u, v] \in E$ iff there is a disk T such that $u, v \in \text{bd } T$ ⁴ and $S \cap \text{int } T = \emptyset$. Finally, a plane graph $G = (S, E)$ is named a *nearest neighbour* graph when $[u, v] \in E$ iff either u or v is the nearest Euclidean neighbour for v or u respectively.

Hardness Results. Our first result claims strong NP-hardness of IPGD within the class of Delaunay triangulations and some known classes of their connected subgraphs (Gabriel and relative neighbourhood graphs) for $r \in [d_{\min}, d_{\max}]$ and $\mu = \frac{d_{\max}}{d_{\min}} = O(|S|)$. IPGD remains strongly NP-hard within the class of nearest neighbour graphs for $r \in [d_{\max}, \eta d_{\max}]$ with a large constant η and $\mu \leq 4$. Furthermore, we have the same NP-hardness results under the same restrictions on r and μ even if we are bound to choose points of C close to vertices of G . The upper bound on μ for Delaunay triangulations is comparable with the lower bound $\mu = \Omega(\sqrt[3]{n^2})$ which holds true with positive probability for Delaunay triangulations produced by n random independent points on the unit disk [2]. Thus, declared restrictions on r and μ define natural instances of IPGD.

An upper bound on μ implies an upper bound on the ratio of the largest and smallest aspect ratio of objects from $\mathcal{N}_r(E)$. The HITTING SET problem is generally easier when sets from \mathcal{N} have almost equal aspect ratio bounded from above by some constant. Our result for the class of nearest neighbour graphs gives the problem NP-hardness in the case where objects of $\mathcal{N}_r(E)$ have almost equal constant aspect ratio.

In distinction to known results for the HITTING SET problem mentioned above our study is mostly for its continuous setting with the structured system $\mathcal{N}_r(E)$ formed by an edge set of a specific plane graph; each set from $\mathcal{N}_r(E)$ is of the special form of Minkowski sum of some graph edge and radius r disk. Our proofs are elaborate complexity reductions from the CDC problem which is intimately related to IPGD.

² $\text{int } N$ is the set of interior points of N .

³ When U coincides with some prescribed finite set.

⁴ $\text{bd } T$ denotes the set of boundary points of T .

Positive Results. Let $R(E)$ be the smallest radius of the disk that intersects all segments from the edge set E . As opposed to the cases where either $r \in [d_{\min}, d_{\max}]$ or $r \in [d_{\max}, \eta d_{\max}]$, IPGD is solvable within the class of simple plane graphs, for which the inequality $r \geq \eta R(E)$ holds uniformly for some fixed $\eta > 0$, in $O(k^2|E|^{2k+1})$ time with $k = \left\lceil \frac{\sqrt{2}}{\eta} \right\rceil^2$. Above inequality implies an upper bound k on its optimum. Taking proof of $W[1]$ -hardness into account of parameterized version of CDC [10] as well as the reduction used to prove the Theorem 2 of this paper, it seems unlikely to improve this time bound to $O(f(k)|E|^c)$ for any computable function f and any constant $c > 0$.

Finally, we present an $8p(1+2\lambda)$ -approximation $O(|E| \log |E|)$ -time algorithm for IPGD when the inequality $r \geq \frac{d_{\max}}{2\lambda}$ holds true uniformly within a class of simple plane graphs for a constant $\lambda > 0$, where $p(x)$ is the smallest number of unit disks needed to cover any disk of radius $x > 1$. It corresponds to the case where segments from E have their lengths uniformly bounded from above by some linear function of r , or, in other words, when objects from $\mathcal{N}_r(E)$ have their aspect ratio bounded from above by $1+\lambda$. A similar but more complex $O(|E|^{1+\varepsilon})$ -time constant factor approximation algorithm is given in [7] to approximate the HITTING SET problem for sets of objects whose aspect ratio is bounded from above by some constant.

2 NP-Hardness Results

We give complexity analysis for the IPGD problem by considering its setting where $r \in [d_{\min}, d_{\max}]$. Under this restriction on r IPGD coincides neither with known VERTEX COVER problem nor with CDC. In fact it is equivalent (see the Introduction) to the geometric HITTING SET problem for the set $\mathcal{N}_r(E)$ of Euclidean r -neighbourhoods of edges of G . For the IPGD problem we claim its NP-hardness even if we restrict the graph G to be either a Delaunay triangulation or some of its known subgraphs. We keep the ratio $\mu = \frac{d_{\max}}{d_{\min}}$ bounded from above, thus, imposing an upper bound on the ratio of the largest and smallest aspect ratio of objects from $\mathcal{N}_r(E)$. We show that IPGD remains intractable even in its simple case where $r = \Theta(d_{\max})$ and μ is bounded by some small constant or, equivalently, when objects of $\mathcal{N}_r(E)$ have close constant aspect ratio.

Our first hardness result for IPGD is obtained by using a complexity reduction from the CDC problem. Below we describe a class of hard instances of the CDC problem which correspond to hard instances of the IPGD problem for Delaunay triangulations with relatively small upper bound on the parameter μ .

2.1 NP-Hardness of the CDC Problem

To single out the class of hard instances of the CDC problem a reduction is used in [9] from the strongly NP-complete minimum dominating set problem which is formulated as follows: given a simple planar graph $G_0 = (V_0, E_0)$ of degree at most 3, find the smallest cardinality set $V'_0 \subseteq V_0$ such that for each $u \in V_0 \setminus V'_0$ there is some $v = v(u) \in V'_0$ which is adjacent to u .

Below an integer grid denotes the set of all points on the plane with integer-valued coordinates each of which belongs to some bounded interval. An *orthogonal* drawing of the graph G_0 on some integer grid is the drawing whose vertices are represented by points on that grid whereas its edges are given in the form of polylines that are composed of connected axis-parallel straight line segments of the form $[p_1, p_2], [p_2, p_3], \dots, [p_{k-1}, p_k]$, and intersecting only at the edge endpoints p_1 and p_k , where each point p_i again belongs to the grid. In [9] strong NP-hardness of CDC is proved by reduction from the minimum dominating set problem. This reduction involves using plane orthogonal drawing of G_0 on some integer grid. More specifically, a set D is build on that grid with $V_0 \subset D$. The resulting hard instance of the CDC problem is for the set D and some integer (constant) radius $r_0 \geq 1$. Let us observe that G_0 admits an orthogonal drawing (Theorem 1 [15]) on the grid of size $O(|V_0|) \times O(|V_0|)$ whereas total length of each its edge is of the order $O(|V_0|)$. Proof of strong NP-hardness of CDC could be conducted taking into account this observation. We can formulate (see Theorems 1 and 3 from [9]).

Theorem 1 [9]. *The CDC problem is strongly NP-hard for a constant integer radius r_0 and point sets D on the integer grid of size $O(|D|) \times O(|D|)$. It remains strongly NP-hard even if we restrict centers of radius r_0 disks to be at the points of D .*

Remark 1. For every simple planar graph G_0 of degree at most 3 its orthogonal drawing can be constructed such that at least one of its edges is a polyline which is composed of at least two axis-parallel segments.

2.2 NP-Hardness of the IPGD Problem for Delaunay Triangulations

To build a reduction from the CDC problem on the set D (as constructed in proof of the Theorem 1 from [9]), we exploit a simple idea that a radius r disk covers a set of points $D' \subset D$ iff a slightly larger disk intersects (and, sometimes, covers) straight line segments, each of which is close to some point of D' and has a small length with respect to distances between points of D . Then a proximity graph H is build whose vertex set coincides with the set of endpoints of small segments corresponding to points of D . Since H usually contains these small segments as its edges, this technique gives NP-hardness for the IPGD problem within numerous classes of proximity graphs. The following technical lemma holds which reports an r -dependent lower bound on the distance between any point with integer coordinates and a radius r circle through the pair of integer-valued points.

Lemma 1. *Let $X \subset \mathbb{Z}^2$, $r \geq 1$ is an integer, $\rho(u; v, w)$ denotes the minimum of two Euclidean distances from an arbitrary point $u \in X$ to the union of two radius r circles which pass through distinct points v and w from X , where $|v - w|_2 \leq 2r$, \mathbb{Z} is the set of integers and $|\cdot|_2$ is Euclidean norm. Then*

$$\min_{u \notin C(v, w), v \neq w, u, v, w \in X, |v - w|_2 \leq 2r} \rho(u; v, w) \geq \frac{1}{480r^5},$$

where $C(v, w)$ is the union of two radius r circles passing through v and w .

Let us formulate the following restricted form of IPGD.

VERTEX RESTRICTED IPGD (VRIPGD(δ)): given a simple plane graph $G = (V, E)$, a constant $\delta > 0$ and a constant $r > 0$, find the least cardinality set $C \subset \mathbb{R}^2$ such that each $e \in E$ is within Euclidean distance r from some point $c = c(e) \in C$ and $C \subset \bigcup_{v \in V} B_\delta(v)$.

Theorem 2. *Both IPGD and VRIPGD(δ) problems are strongly NP-hard for $r \in [d_{\min}, d_{\max}]$, $\mu = O(n)$ and $\delta = \Theta(r)$ within the class of Delaunay triangulations, where n is the number of vertices in triangulation.*

Proof. Let us prove that IPGD is strongly NP-hard. Proof technique for the VRIPGD(δ) problem is analogous taking into account the Theorem 1 (see also proof of the Theorem 3 from [9] for details). For any hard instance of the CDC problem, which the Theorem 1 reports, the IPGD problem instance is built for $r = r_0 + \delta$ and $\delta = \frac{1}{2000^2 2r_0^{11}}$ as follows. For every $u \in D$ points u_0 and v_0 are found such that $|u - u_0|_\infty \leq \delta/2$ and $|u - v_0|_\infty \leq \delta/2$, where $I_u = [u_0, v_0]$ has Euclidean length at least $\delta/2$ and $|\cdot|_\infty$ denotes norm in \mathbb{R}^2 equal to the maximum of absolute values of vector coordinates. More specifically, let us set $I_D = \{I_u = [u_0, v_0] : u \in D\}$. Endpoints of segments from I_D are constructed in sequential manner in polynomial time and space by defining a new segment I_u to provide general position for the set of endpoints of the set $I_{D'} \cup \{I_u\}$, $D' \subset D$, where segments of $I_{D'}$ are already defined. Here endpoints of I_u are chosen in the rational grid that contains u whose elementary cell size is $\frac{c_1}{|D|^2} \times \frac{c_1}{|D|^2}$ for some small absolute rational constant $c_1 = c_1(\delta)$. Assuming $u = (u_x, u_y)$, the point u_0 is chosen in the lower part of the grid with y -coordinates less than $u_y - \delta/4$ whereas v_0 is taken from the upper one for which y -coordinates exceed $u_y + \delta/4$.

Let S be the set of endpoints of segments from I_D . Every disk having I_u as its diameter does not contain any points of S distinct from endpoints of I_u . Let $G = (S, E)$ be a Delaunay triangulation for S which can be computed in polynomial time and space in $|D|$. Obviously, each segment I_u coincides with some edge from E . We have $d_{\min} \leq r$ and $\mu = O(|S|)$. It remains to prove that $r \leq d_{\max}$. Due to the Remark 1 and a construction of the set D (see Fig. 1 and proof of the Theorem 1 from [9]) the set S can be constructed such that the inequality $r \leq d_{\max}$ holds true for G . Moreover, representation length for vertices of S is polynomial with respect to representation length for points of D .

Let k be a positive integer. Obviously, centers of at most k disks of radius r_0 , containing D in their union, give centers of radius $r > r_0$ disks whose union is intersected with each segment from E . Conversely, let T be a disk of radius r which intersects a subset $I_{D'} = \{I_u : u \in D'\}$ of segments for some $D' \subseteq D$. When $|D'| = 1$, it is easy to transform T into a disk which contains the segment $I_{D'}$. Points of D have integer coordinates. Moreover, squared Euclidean distance between each pair of points of the subset D' does not exceed $(2r_0 + 4\delta)^2 = 4r_0^2 + 16r_0\delta + 16\delta^2$. As $r_0 \in \mathbb{Z}$, points from D' are located within the distance $2r_0$ from each other. Let us use Helly theorem. Let R be the minimum radius

of the disk T_0 , containing any triple u_1, u_2 and u_3 from D' . W.l.o.g. we suppose that, say, u_1 and u_2 are on the boundary of T_0 and denote its center by O . Obviously, $R \leq r_0 + 2\delta$. Let us show that the case $R > r_0$ is void. The center of T_0 can be shifted along the midperpendicular to $[u_1, u_2]$ to have u_1 and u_2 at the distance r_0 from the shifted center O' . The distance from the point u_3 to the radius r_0 circle centered at O' does not exceed

$$\begin{aligned} |O - u_3|_2 + |O - O'|_2 - r_0 &\leq 2\delta + \sqrt{(r_0 + 2\delta)^2 - \delta_1^2} - \sqrt{r_0^2 - \delta_1^2} \\ &= 2\delta + \frac{4r_0\delta + 4\delta^2}{\sqrt{(r_0 + 2\delta)^2 - \delta_1^2} + \sqrt{r_0^2 - \delta_1^2}} \leq 2\delta + 2\sqrt{r_0\delta + \delta^2} < \frac{1}{480r_0^5}, \end{aligned}$$

where $\delta_1 = \frac{|u_1 - u_2|_2}{2} \leq r_0$. By the Lemma 1 we have $R \leq r_0$. Thus, D' is contained in some disk of radius r_0 . Given a set of points on the plane, the smallest radius disk can be found in polynomial time and space which covers this set. Therefore we can convert any set of at most k disks of radius r whose union is intersected with each segment from E to some set of at most k disks of radius r_0 whose union covers D .

Using the Corollary 1 of Sect. 4.2 from [2] and the Theorem 1 from [11] we arrive at the lower bound $\mu = \Omega(\sqrt[3]{n^2})$ which holds true with positive probability for Delaunay triangulations produced by n random uniform points on the unit disk. Thus, the order of the parameter μ for the considered class of hard instances of the IPGD problem is comparable with the one for random Delaunay triangulations.

2.3 NP-Hardness of IPGD for Other Classes of Proximity Graphs

The same proof technique could be applied for proving NP-hardness of the problem within the other classes of proximity graphs. Let us start with some definitions. The following graphs are connected subgraphs of Delaunay triangulations. A plane graph $G = (S, E)$ is called a *Gabriel graph* when $[u, v] \in E$ iff the disk having $[u, v]$ as its diameter does not contain any other points of S distinct from u and v . A *relative neighbourhood graph* is the plane graph G with the same vertex set for which $[u, v] \in E$ iff there is no any other point $w \in S$ such that $w \neq u, v$ with $\max\{|u - w|_2, |v - w|_2\} < |u - v|_2$. Finally, a plane graph is called a *minimum Euclidean spanning tree* if it is the minimum weight spanning tree of the weighted complete graph $K_{|S|}$ whose vertices are points of S such that its edge weight is given by Euclidean distance between the edge endpoints.

Corollary 1. *Both IPGD and VRIPGD(δ) problems are strongly NP-hard for $r \in [d_{\min}, d_{\max}]$, $\mu = O(n)$ and $\delta = \Theta(r)$ within classes of Gabriel, relative neighbourhood graphs and minimum Euclidean spanning trees as well as for $r \in [d_{\max}, \eta d_{\max}]$ and $\mu \leq 4$ within the class of nearest neighbour graphs where η is a large constant.*

3 Positive Results

3.1 Polynomial Solvability of the IPGD Problem for Large r

Before presenting polynomially solvable case of the IPGD problem we are to take some preprocessing. It is aimed at reducing the set of points, among which centers of radius r disks are chosen, to a finite set whose cardinality is bounded from above by some polynomial in $|E|$.

Problem Preprocessing. As was mentioned in the Introduction, the IPGD problem coincides with the HITTING SET problem considered for Euclidean r -neighbourhoods of graph edges which form the system denoted by $\mathcal{N}_r(E)$. Their boundaries are composed of four parts: two half-circles and two parallel straight line segments. W.l.o.g. we can assume that intersection of any subset of objects from $\mathcal{N}_r(E)$ (if nonempty) contains a point from the intersection of boundaries of two objects from $\mathcal{N}_r(E)$. Thus, our choice of points to form a feasible solution to the IPGD problem can be restricted to the set of intersection points of boundaries of pairs of objects from $\mathcal{N}_r(E)$. The following lemma can be considered a folklore.

Lemma 2. *Let $G = (V, E)$ be a simple plane graph. Each feasible solution C to the IPGD problem for G can be converted in polynomial time and space (in $|E|$) to a feasible solution $D \subset D_r(G)$ to IPGD for G with $|D| \leq |C|$, where $D_r(G) \subset \mathbb{R}^2$ is some set of cardinality of the order $O(|E|^2)$ which can be constructed in polynomial time and space.*

Polynomially Solvable Case of IPGD. In distinction to the cases where either $r \in [d_{\min}, d_{\max}]$ or $r = \Theta(d_{\max})$ the IPGD problem is polynomially solvable for $r = \Omega(R(E))$, where $R(E)$ is the smallest radius of the disk that intersects all segments from E . Due to [3] the IPGD problem is solvable in $O(|E|)$ time within the class of plane graphs for which the inequality $r \geq R(E)$ holds uniformly.

Let us consider the IPGD problem within the class of plane graphs for which the inequality $r \geq \eta R(E)$ holds uniformly for some fixed constant $0 < \eta < 1$. Since every radius r disk contains an axis-parallel rectangle whose side is equal to $r\sqrt{2}$, roughly at most $\left\lceil \frac{\sqrt{2}R(E)}{r} \right\rceil^2 \leq \left\lceil \frac{\sqrt{2}}{\eta} \right\rceil^2 = k(\eta) = k$ radius r disks are needed to intersect all segments from E . Therefore the brute-force search algorithm could be applied that just sequentially tries each subset of $D_r(G)$ of cardinality at most k . This amounts roughly to $O(k^2|E|^{2k+1})$ time complexity. Thus, we arrive at the polynomial time algorithm whose complexity depends exponentially on $1/\eta$. This algorithm gives an optimal solution to the IPGD problem taking the Lemma 2 into account.

3.2 Approximation Algorithm for the IPGD Problem

Below the approximation algorithm is reported for the IPGD problem whose approximation factor depends on the maximum aspect ratio among objects of

$\mathcal{N}_r(E)$. More specifically, let us focus on the case of IPGD where the inequality $r \geq \frac{d_{\max}}{2\lambda}$ holds uniformly within some class \mathcal{G}_λ of simple plane graphs for a constant $\lambda > 0$. It corresponds to the situation where objects from the system $\mathcal{N}_r(E)$ have their aspect ratio bounded from above by $1 + \lambda$. In this case it turns out that the problem admits an $O(1)$ -approximation algorithm whose factor depends on λ . The following auxiliary problem is considered to formulate it.

COVER ENDPOINTS OF SEGMENTS WITH DISKS (CESD). Let $S(G) \subseteq V$ be the set of endpoints of edges of G . It is required to find the smallest cardinality set of radius r disks whose union contains $S(G)$.

ALGORITHM. Compute and output 8-approximate solution to the CESD problem using $O(|E| \log OPT_{CESD}(S(G), r))$ -time algorithm (see Sects. 2 and 4 from [8]).

We call a subset $V' \subseteq V$ by a *vertex cover* for $G = (V, E)$ when $e \cap V' \neq \emptyset$ for any $e \in E$. The statement below bounds the ratio of optima for CESD and IPGD problems in the general case where $S(G)$ is an arbitrary vertex cover of the graph G .

Statement 3. *The following bound holds true for any graph $G \in \mathcal{G}_\lambda$ without isolated vertices:*

$$\frac{OPT_{CESD}(S(G), r)}{OPT_{IPGD}(G, r)} \leq p(1 + 2\lambda)$$

where $p(x)$ is the smallest number of unit disks needed to cover radius x disk.

Proof. Let $C_0 = C_0(G, r) \subset \mathbb{R}^2$ be an optimal solution to IPGD for a given $G \in \mathcal{G}_\lambda$. Set $E(c, G) := \{e \in E : c \in N_r(e)\}$, $c \in C_0$. For every $e \in E(c, G)$ there is a point $c(e) \in e$ with $|c - c(e)|_2 \leq r$. Any point from the set $S(c, G)$ of endpoints of segments from $E(c, G)$ is within the distance $r + d_{\max}$ from the point c . Due to definition of p , at most $p(1 + 2\lambda)$ radius r disks are needed to cover radius $r + d_{\max}$ disk. Therefore the set $S(G) \subseteq \bigcup_{c \in C_0} S(c, G)$ is contained in the union of at most $|C_0|p(1 + 2\lambda)$ radius r disks.

Corollary 2. *The algorithm is $8p(1 + 2\lambda)$ -approximate.*

Remark 2. Approximation factor of the algorithm is in fact lower when \mathcal{G}_λ is the subclass of Delaunay triangulations or of their subgraphs. Indeed, in this case there is no need to cover the whole radius $r + d_{\max}$ disk with radius r disks.

Remark 3. If $S(G)$ is the set of midpoints of segments from E , the algorithm is $8p(1 + \lambda)$ -approximate.

4 Conclusion

Complexity and approximability are studied for the problem of intersecting a structured set of straight line segments with the smallest number of disks of radii $r > 0$ where a structural information about segments is given in the form

of an edge set of a plane graph. It is shown that the problem is strongly NP-hard within the class of Delaunay triangulations and some of their subgraphs for small and medium values of r while for large r it is polynomially solvable. Fast approximation algorithm is given for the IPGD problem whose approximation factor depends on the maximum aspect ratio among objects from $\mathcal{N}_r(E)$. Of course, those algorithms are of particular interest whose factor is bounded from above by some absolute constant. This sort of algorithms is our special focus for future research.

A Proof of the Lemma 1

Proof. Let $u = (x, y)$, $v = (x_1, y_1)$ and $w = (x_2, y_2)$ be distinct points of X . Consider an arbitrary radius r circle (out of two circles) which passes through v and w , and denote its center by O . A lower bound is obtained below for the distance $\pi = \pi(u; v, w)$ from that circle to the point $u \notin C(v, w)$.

Let $\Delta = |v - w|_2$, $\lambda = \sqrt{r^2 - \frac{\Delta^2}{4}}$, $a = (u - v, u - w)$ and $b = (u - v, (v - w)^\perp)$, where $(v - w)^\perp = \pm(y_1 - y_2, -x_1 + x_2)$. The distance $\pi > 0$ can be written in the form:

$$\pi = \pi(u; v, w) = \left\| \frac{v + w}{2} - \lambda \frac{(v - w)^\perp}{|v - w|_2} - u \right\|_2 - r = \left\| \frac{a + \frac{2\lambda b}{\Delta}}{\sqrt{a + \frac{2\lambda b}{\Delta} + r^2} + r} \right\|.$$

Without loss of generality it is assumed that u is in the $2r$ radius disk centered at O . Indeed, we get $\pi \geq r \geq \frac{1}{r}$ otherwise. Let us bound denominator of the fraction π , taking into account that $\Delta \leq 2r$, $|u - v|_2 \leq |u - O|_2 + |O - v|_2 \leq 3r$ and $|b|/\Delta \leq 3r$:

$$\sqrt{a + \frac{2\lambda b}{\Delta} + r^2} + r \leq 5r.$$

As points of X have integer coordinates, a and b are integers. For $\Delta^2 = 4r^2$ we get $\pi \geq \frac{1}{5r}$. When $\Delta^2 \leq 4r^2 - 1$, it is enough to prove the inequality

$$\left| a + \frac{2\lambda b}{\Delta} \right| \geq \frac{1}{96r^4}. \quad (1)$$

Indeed, again, combining this bound with the aforementioned upper bound for denominator of the fraction π , we get $\pi \geq \frac{1}{480r^5}$.

For integer $\frac{2\lambda b}{\Delta}$ the left-hand side of the inequality (1) is at least 1. Thus, it remains for us to prove the inequality (1) for the case where $\frac{2\lambda b}{\Delta} \notin \mathbb{Z}$. Suppose that $q = \left\{ \frac{2\lambda b}{\Delta} \right\} > 0$ and $k = \left[\frac{2\lambda b}{\Delta} \right]$, where $\{\cdot\}$ and $[\cdot]$ denote fractional and integer part of real number respectively. In fact, the term $\min\{q, 1 - q\}$ can be bounded from below. Let us start estimating with q . First, it is assumed that $\gamma = \frac{4r^2b^2}{\Delta^2} \in \mathbb{Z}$. We have $k^2 < \frac{4\lambda^2b^2}{\Delta^2} < (k+1)^2$. As $q > 0$, we get $q \geq \left\{ \sqrt{k^2 + 1} \right\}$. Due to concavity of the square root we have

$$\left\{ \sqrt{k^2 + 1} \right\} = \left\{ \sqrt{\frac{2k \cdot k^2}{2k+1} + \frac{(k+1)^2}{2k+1}} \right\} \geq \left\{ k + \frac{1}{2k+1} \right\} = \frac{1}{2k+1}$$

$$\geq \frac{1}{\frac{4\lambda|b|}{\Delta} + 1} \geq \frac{1}{13r^2}.$$

Now the case is considered where $\gamma \notin \mathbb{Z}$. As $2kq + q^2 \geq \{2kq + q^2\} = \{\gamma\}$, we have that

$$q \geq \sqrt{k^2 + \{\gamma\}} - k \geq \frac{\{\gamma\}}{\sqrt{k^2 + \{\gamma\}} + k} \geq \frac{\frac{1}{\Delta^2}}{\frac{4r|b|}{\Delta}} \geq \frac{1}{12r^2\Delta^2} \geq \frac{1}{48r^4}.$$

Let us get a lower bound for $1 - q$. Again, assume that $\gamma \in \mathbb{Z}$. Arguing analogously, we arrive at the bound

$$2k(1 - q) + (1 - q)^2 \geq \{(k + 1 - q)^2\} = \left\{ (k + 1)^2 - \frac{4\lambda^2 b^2}{\Delta^2} - 2q(1 - q) \right\} \geq \frac{1}{2}.$$

Resolving the quadratic inequality with respect to $1 - q$, we get:

$$1 - q \geq \sqrt{k^2 + \frac{1}{2}} - k = \frac{\frac{1}{2}}{\sqrt{k^2 + \frac{1}{2}} + k} \geq \frac{1}{\frac{8r|b|}{\Delta}} \geq \frac{1}{24r^2}.$$

Now let $\gamma \notin \mathbb{Z}$. Let us consider the subcase, where $\{\gamma\} + 2q(1 - q) > 1$. We get

$$\left\{ (k + 1)^2 - \frac{4\lambda^2 b^2}{\Delta^2} - 2q(1 - q) \right\} \geq 1 - \{\gamma\} \geq \frac{1}{\Delta^2}.$$

Resolving the corresponding inequality with respect to $1 - q$, we arrive at the analogous lower bound $1 - q \geq \frac{1}{48r^4}$.

Now we are to address the case where $\{\gamma\} + 2q(1 - q) < 1$. Obviously,

$$\left\{ (k + 1)^2 - \frac{4\lambda^2 b^2}{\Delta^2} - 2q(1 - q) \right\} = 1 - \{\gamma\} - 2q(1 - q).$$

For $1 - q < \frac{1}{4\Delta^2}$ we have $1 - \{\gamma\} - 2q(1 - q) \geq \frac{1}{\Delta^2} - \frac{1}{2\Delta^2} = \frac{1}{2\Delta^2}$. Arguing analogously, we obtain the following bound $1 - q \geq \frac{1}{96r^4}$; otherwise, we get $1 - q \geq \frac{1}{4\Delta^2} \geq \frac{1}{16r^2}$.

For $\{\gamma\} + 2q(1 - q) = 1$ we have:

$$1 - q = \frac{1 - \{\gamma\}}{2q} \geq \frac{1 - \{\gamma\}}{2} \geq \frac{1}{2\Delta^2} \geq \frac{1}{8r^2}.$$

Finally, we arrive at the claimed bound:

$$\min\{q, 1 - q\} \geq \frac{1}{96r^4}.$$

References

1. Agarwal, P.K., Efrat, A., Ganjugunte, S.K., Hay, D., Sankararaman, S., Zussman, G.: The resilience of WDM networks to probabilistic geographical failures. *IEEE/ACM Trans. Netw.* **21**(5), 1525–1538 (2013)
2. Arkin, E.M., Anta, A.F., Mitchell, J.S., Mosteiro, M.A.: Probabilistic bounds on the length of a longest edge in Delaunay graphs of random points in d dimensions. *Comput. Geom.* **48**(2), 134–146 (2015)
3. Bhattacharya, B.K., Jadhav, S., Mukhopadhyay, A., Robert, J.M.: Optimal algorithms for some intersection radius problems. *Computing* **52**(3), 269–279 (1994)
4. Bose, P., De Carufel, J.-L., Durocher, S., Taslakian, P.: Competitive online routing on delaunay triangulations. In: Ravi, R., Gørtz, I.L. (eds.) SWAT 2014. LNCS, vol. 8503, pp. 98–109. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08404-6_9
5. Bose, P., Kirkpatrick, D.G., Li, Z.: Worst-case-optimal algorithms for guarding planar graphs and polyhedral surfaces. *Comput. Geom.* **26**(3), 209–219 (2003)
6. Chan, T.M., Grant, E.: Exact algorithms and APX-hardness results for geometric packing and covering problems. *Comput. Geom.* **47**(2), 112–124 (2014)
7. Efrat, A., Katz, M.J., Nielsen, F., Sharir, M.: Dynamic data structures for fat objects and their applications. *Comput. Geom.* **15**, 215–227 (2000)
8. Gonzalez, T.F.: Covering a set of points in multidimensional space. *Inf. Process. Lett.* **40**(4), 181–188 (1991)
9. Hasegawa, T., Masuyama, S., Ibaraki, T.: Computational complexity of the m -center problems on the plane. *Trans. Inst. Electron. Commun. Eng. Japan Sect. E* **64**(2), 57–64 (1981)
10. Marx, D.: Efficient approximation schemes for geometric problems? In: Brodal, G.S., Leonardi, S. (eds.) ESA 2005. LNCS, vol. 3669, pp. 448–459. Springer, Heidelberg (2005). https://doi.org/10.1007/11561071_41
11. Onoyama, T., Sibuya, M., Tanaka, H.: Limit distribution of the minimum distance between independent and identically distributed d -dimensional random variables. In: de Oliveira, J.T. (ed.) Statistical Extremes and Applications. NATO ASI Series (Series C: Mathematical and Physical Sciences), vol. 131, pp. 549–562. Springer, Dordrecht (1984). https://doi.org/10.1007/978-94-017-3069-3_42
12. O'Rourke, J.: Art Gallery Theorems and Algorithms. Oxford University Press, New York (1987)
13. Har-Peled, S., Quanrud, K.: Approximation algorithms for polynomial-expansion and low-density graphs. In: Bansal, N., Finocchi, I. (eds.) ESA 2015. LNCS, vol. 9294, pp. 717–728. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48350-3_60
14. Stojmenovic, I., Urrutia, J., Bose, P., Morin, P.: Routing with guaranteed delivery in ad hoc wireless networks. *Wirel. Netw.* **7**(6), 609–616 (2001)
15. Tamassia, R., Tollis, I.G.: Planar grid embedding in linear time. *IEEE Trans. Circ. Syst.* **36**, 1230–1234 (1989)

Analysis of Dynamic Behavior Through Event Data

On the Efficient Application of Aho-Corasick Algorithm in Process Mining

Andrey M. Konchagin and Anna A. Kalenkova^(✉)

National Research University Higher School of Economics, Moscow, Russia
amkonchagin@hse.ru, akalenkova@edu.hse.ru

Abstract. In this paper we present an approach for searching sub-traces in event logs, generated by information systems. Our technique is heavily based on the Aho-Corasick algorithm, and extends it with simultaneous search on several event log traces. The computational complexity of the proposed approach was estimated. Moreover, the approach was implemented and verified on real-life event logs. It was shown that it allows to reduce the search time for event logs with a high proportion of similar traces.

1 Introduction

Process mining [1] is a research field focused on the analysis of systems' behavior recorded in a form of event logs. Process mining offers dozens of methods for process model discovery. One of the challenging research areas within the discovery is mining repeating common structures, which can be considered as sub-processes. Techniques for discovering repeating sub-traces from event logs were proposed and described in [2,3]. In this work we start from the belief that not all of the automatically discovered patterns are meaningful, and in some cases it is more feasible to use an expert defined dictionary of sub-traces. Having such a dictionary allows to find all occurrences of sub-traces, and thus, structure and reduce the entire event log, making its further analysis easier. Thus, this approach can be further used in discovering process models, containing sub-processes [4,5]. Especially it can show its effectiveness when applied to event logs containing high proportion of similar repeating sub-traces.

In order to find all occurrences of sub-traces in the event log we will utilize the Aho-Corasick algorithm [6]. In contrast to [2] we will not analyze traces sequentially, but will propose an effective extension of the classical approach [6], which analyzes several event log traces simultaneously. For that purpose the initial event log will be presented as a prefix tree, and a novel algorithm for replaying this prefix tree on a finite state machine, constructed from the pre-defined dictionary, will be presented. The proposed algorithm was implemented and tested on a real-life event log data. The effectiveness of the approach was both theoretically and practically estimated. Another important advantage of the proposed technique in comparison to other string matching algorithms [7,8] is that it is based on the application of a finite state machine. Thus, this method

can be further extended to handle more advanced templates relying on regular expressions and compositions of finite state machines.

2 Preliminaries

This section contains basic notions, which will be referred later in the text. $\mathcal{P}(X)$ denotes the set of all multisets over set X . By $p = [x_1^3, x_2^2]$, where $p \in \mathcal{P}(X)$, we denote that x_1 and x_2 appear in p three and two times. For set X , X^+ is the set of non-empty finite sequences over X .

Let \mathcal{A} be a set of activities. Trace $\sigma \in \mathcal{A}^+$ is a non-empty sequence of activities. Let $\sigma = \langle a_1, a_2, \dots, a_n \rangle$ be a trace. Sequence $\langle a_i, a_{i+1}, \dots, a_j \rangle$, such that $i, j \in \overline{1, k}$ and $i \leq j$, is a *sub-trace* of σ . By $\sigma_k = \langle a_1, a_2, \dots, a_k \rangle$ we will denote a sub-trace called *prefix* of σ of the length k , and $\sigma(i) = a_i$, $1 \leq i \leq n$, stands for the activity in i th position of the trace. $|\sigma|$ denotes the total length of the trace σ .

Event log L is a multiset over a set of traces, i.e., $L \in \mathcal{P}(\mathcal{A}^+)$. A *finite state machine* is a tuple $FSM = (S, E, T, s_i, S_f)$, where S is a finite set of *states*, E is a finite set of *events* (or *activities*), $T \subseteq (S \times E \times S)$ is a set of *transitions*, $s_i \in S$ is an *initial state*, and $S_f \subseteq S$ is a set of *accepting states*. Trace $t = \langle a_1, \dots, a_n \rangle \in \mathcal{A}^+$ is *accepted* by a finite state machine $FSM = (S, \mathcal{A}, T, s_i, S_f)$ iff there exists a sequence of transitions $(s_i, a_1, s_1), (s_1, a_2, s_2), \dots, (s_{n-1}, a_n, s_n)$, leading from the initial state s_i to an accepting state $s_n \in S_f$.

A *prefix tree* for an event log L is a finite state machine $Pref_L = (S, \mathcal{A}, T, s_i, S_f)$, such that $S = \{\forall \sigma \in L, \forall k \in [1, |\sigma|] | \sigma_k\} \cup \{\langle \rangle\}$, $s_i = \langle \rangle$, i.e., the set of all prefixes of the traces unified with an empty sequence, $T = \{\forall \sigma \in L : |\sigma| \geq 2, \forall k \in [1, |\sigma| - 1] | (\sigma_k, \sigma(k + 1), \sigma_{k+1})\} \cup \{\forall \sigma \in L | (s_i, \sigma(1), \sigma_1)\}$, $S_f = \{\forall \sigma \in L | \sigma_{|\sigma|}\}$.

To model a dictionary of sub-traces a special type of a finite state machine should be introduced. A *dictionary* is a finite state machine $D = (S, \mathcal{A} \cup \{\epsilon, \delta\}, T, s_i, S_f)$ with events $\epsilon, \delta \notin \mathcal{A}$, which mark special transitions, guiding the search of sub-traces.

3 Motivating Example

Let us consider an event log over set $\mathcal{A} = \{a, b, c, d, e, f\}$:

$$L = [\langle a, b, d, e \rangle^{101}, \langle a, b, d, c \rangle^{78}, \langle b, c, a, b, d \rangle^{43}, \langle b, c, f, b, c \rangle^{37}, \langle b, c, b, c, a, b, d \rangle^5, \langle b, c, b \rangle^4].$$

A prefix tree constructed for this event log is shown in Fig. 1a. Final states of this tree are highlighted in white. A dictionary defined on the basis of a predefined sub-traces set $\{\langle a, b, d \rangle, \langle b, c, a, d \rangle, \langle b, c, f \rangle, \langle c, f \rangle\}$ is shown in Fig. 1b. According to the Aho-Corasick algorithm [6] this dictionary is used to find sub-strings through replaying the initial string. If the accepting state of the dictionary is reached, then the sub-trace is found, and the search starts again from the initial state of the dictionary, considering the input string from the next symbol. Besides

that, the algorithm uses additional transitions, which optimize the search time. These are so-called *prefix* and *dictionary links* (denoted by ϵ and δ respectively). Returning back to our example, trace $\sigma = \langle b, c, b, c, a, b, d \rangle$ fails to be replayed on the symbol $\sigma(3) = b$ in the state $\langle b, c \rangle$ of the dictionary. In that case a prefix link is executed and the current state changes to $\langle c \rangle$. Thus, prefix links allow not to start the matching procedure again from the next symbol of the trace, but to consider a maximum sub-trace of symbols that still can be replayed by a dictionary (in our case this sub-trace is $\langle c \rangle$). In Fig. 1b. it is assumed that all the states without outgoing prefix links have prefix links leading to the root of the dictionary. Note that after $\langle b, c, b, c, a \rangle$ is replayed, a prefix link to the state $\langle a \rangle$ will be executed, and finally sub-trace $\langle a, b, d \rangle$ will be identified. Dictionary link do not influence the state machine execution, but help to find nested sub-traces. Thus, in our example finding the entry of sub-trace $\langle b, c, f \rangle$ implies that sub-trace $\langle c, f \rangle$ is also found. That is provided by $(\langle b, c, f \rangle, \delta, \langle c, f \rangle)$ transition.

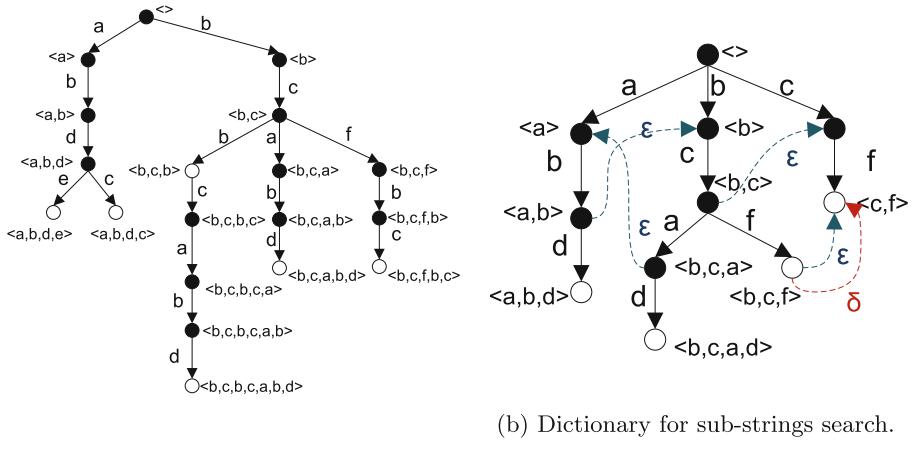


Fig. 1. Prefix tree and a dictionary for sub-strings search

Some traces appear in the event log L multiple times, some of them have identical prefixes. Thus, replaying traces sequentially is not an optimal way to find certain sub-traces. In this work we represent event log in a form of prefix trees and replay these trees on a dictionary in order to find sub-traces in a more optimal way.

4 Algorithm Description

In this section we will describe the proposed algorithm and estimate its computational complexity.

Algorithm 1. Finding sub-traces in an event log

Input: A prefix tree $Pref_L = (S, \mathcal{A}, T, s_i, S_f)$, a dictionary $D = (S', \mathcal{A} \cup \{\epsilon, \delta\}, T', s'_i, S'_f)$.

Output: *Matches*.

```

1: function FIND( $Pref_L, D$ )
2:    $CurrentTreeStack \leftarrow s_i$ ; // stack of the prefix tree states;
3:    $CurrentDStack \leftarrow s'_i$ ; // stack of the dictionary states;
4:    $CurrentState := s_i$ ;  $CurrentDState := s'_i$ ;
5:    $CurrentStateNumber := 0$ ; // current depth;
6:    $forward := \text{true}$ ; // whether we move forward (down) the prefix tree;
7:    $CurrentEvent := \epsilon$ ;
8:   while  $CurrentTreeStack$  not empty do
9:     if ( $forward$  and  $CurrentStateNumber \geq 1$ ) then
10:      // Define ancestors with paths to the current state without branches;
11:      if number of  $CurrentState.Parent.Son == 1$  then
12:         $CurrentState.LinearAnc := CurrentState.Parent.LinearAnc$ ;
13:         $CurrentState.Tempalte \leftarrow CurrentState.Parent.Template$ ;
14:      else
15:         $CurrentState.LinearAnc := CurrentState.Parent$ ;
16:         $CurrentState.Tempalte := \{\}$ ;
17:      end if
18:      // Perform a step within the dictionary by  $CurrentEvent$ ;
19:       $CurrentState.Template, CurrentDState \leftarrow$ 
20:      StepAhoCorasick( $CurrentEvent, CurrentDState$ );
21:      // If current state is an accepting state, define matches for corresponding case ids;
22:      if  $CurrentState \in S_f$  then
23:         $NextState := CurrentState$ ;
24:        while  $NextState \neq s_i$  do
25:           $Matches[\text{Case IDs of } CurrentState] \leftarrow (NextState,$ 
26:           $NextState.Template)$ ;
27:           $NextState := NextState.LinearAnc$ ;
28:        end while
29:      end if
30:       $forward := \text{false}$ ;
31:      for elem in  $CurrentState.Son$  do
32:        if elem is not visited then
33:          // If there is a child state, which has not been considered yet;
34:          if  $CurrentStateNumber \geq 1$  then
35:             $CurrentTreeStack \leftarrow CurrentState$ ;
36:             $CurrentDStack \leftarrow CurrentDState$ ;
37:          end if
38:           $CurrentEvent :=$  event marking transition from  $CurrentState$  to elem;
39:           $CurrentState := elem$ ;
40:           $CurrentStateNumber := CurrentStateNumber + 1$ ;
41:           $forward := \text{true}$ ;
42:          break;
43:        end if
44:      end for
45:      if not  $forward$  then
46:        // If all child nodes were considered, we move up the tree;
47:        mark  $CurrentState$  as visited;
48:         $CurrentStateNumber := CurrentStateNumber - 1$ ;
49:         $CurrentTreeStack \rightarrow CurrentState$ ;
50:         $CurrentDStack \rightarrow CurrentDState$ ;
51:      end if
52:    end while
53:  end function

```

This algorithm will traverse trough a prefix tree and a dictionary simultaneously. A traversal of a prefix tree will be based on the depth-first search algorithm. Current states of a prefix tree and a dictionary will be stored in corresponding stacks: $CurrentTreeStack$ and $CurrentDStack$. The value of the boolean variable $forward$ will define the current direction of a traversal (down

or up the prefix tree). *Matches* is an array, specifying for each trace identifier a set of found matches represented as pairs (a position number in the trace and a corresponding sub-trace). The traversal algorithm calls the Aho-Corasick search procedure from each state of the tree and saves matching templates in *CurrentState.Template* (lines 11–12). Moreover, each state inherits all the templates applied to its maximal ancestor, from which there is a path without branchings (line 6). This allows to optimize the procedure of collecting templates, which can be applied to a current trace (line 13). At the same time a state does not inherit templates of its parents in case of branchings, because this may lead to the duplication of templates by the states with several child nodes. When an accepting state is reached all the corresponding templates are collected, recursively traversing maximal linear ancestors (lines 13–20).

Let N be the total length of the input text, D – the overall length of all words in the dictionary, and A – the size of alphabet. The computational complexity of the Aho-Corasick algorithm is estimated as $O((N + D) \cdot \log A)$ in the worst case. The computational complexity of the proposed approach is $O((V + D) \cdot \log A)$ in the worst case, where V is the number of states in a prefix tree, since this algorithm is based on the depth-first search and incorporates the Aho-Corasick algorithm with V as a total length of the input text.

5 Experimental Results

The proposed optimized approach was implemented and verified on real data. This section presents some results on the analysis of real-life event logs, using the technique proposed. The classical technique, processing each trace individually, and the proposed approach were applied to real-life event logs of a system handling car incidents (*CI*), a banking financial system (*FS*), and a Dutch financial institute data (*FI*). The dictionaries for these event logs were defined manually.

Table 1 shows the results, including the ratio of the most frequent activities left in the log during filtering out infrequent behavior, number of unique traces, total length of traces and dictionaries, and execution times in milliseconds.

According to the results of evaluation, the proposed technique allows to improve the performance of the search algorithm in case of event logs without noise, containing similar sub-traces. If the event log contains noise and infrequent behavior the proposed approach works slower because of switching between prefix tree and dictionary.

Table 1. Application of the classical template search approach and the proposed optimized technique to real-life event logs.

Event log	Ratio of activities	Number of unique traces	Total length of traces	Total length of dictionary	Execution time (not optimized)	Execution time (optimized)
CI	1.0	2278	65533	257	107 ms	596 ms
CI	0.48	455	31740	257	55 ms	41 ms
CI	0.33	58	22053	257	40 ms	25 ms
CI	0.28	34	18593	257	31 ms	18 ms
CI	0.2	13	13364	257	23 ms	13 ms
FS	1.0	4366	262200	1622	513 ms	3183 ms
FS	0.2	113	54275	1622	111 ms	61 ms
FS	0.15	37	40596	1622	80 ms	43 ms
FS	0.13	22	36110	1622	69 ms	37 ms
FI	1.0	4047	561671	1314	1176 ms	1081 ms
FI	0.62	65	348987	1314	508 ms	195 ms
FI	0.45	20	254669	1314	354 ms	137 ms

6 Conclusion

In this paper a novel approach for finding substrings (sub-traces) was proposed. In contrast to the existing Aho-Corasick algorithm it allows for the simultaneous analysis of several strings (traces). The computational complexity of the approach was estimated. This approach was implemented and verified on real-life event logs. It was shown that the efficiency of the approach highly relies on the structure of event logs. Due to the limitations (high dependence on trace prefixes, analysis of sequential sub-traces only) it is planned to improve and extend the approach: use not only prefix trees, but finite state machines of various types (including minimal and communicating finite state machines) in order to identify regular expressions and interleaving sub-traces.

Acknowledgment. This work was supported by the Basic Research Program at the National Research University Higher School of Economics and funded by RFBR and Moscow city Government according to the Research project No 15-37-70008 “mol_a_mos”.

References

1. van der Aalst, W.M.P.: Process Mining: Data Science in Action, 2nd edn. Springer, Heidelberg (2016). <https://doi.org/10.1007/978-3-662-49851-4>
2. Jagadeesh Chandra Bose, R.P., van der Aalst, W.M.P.: Abstractions in process mining: a taxonomy of patterns. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 159–175. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03848-8_12

3. Liesaputra, V., Yongchareon, S., Chaisiri, S.: Efficient process model discovery using maximal pattern mining. In: Motahari-Nezhad, H.R., Recker, J., Weidlich, M. (eds.) BPM 2015. LNCS, vol. 9253, pp. 441–456. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23063-4_29
4. van der Aalst, W.M.P., Kalenkova, A., Rubin, V., Verbeek, E.: Process discovery using localized events. In: Devillers, R., Valmari, A. (eds.) PETRI NETS 2015. LNCS, vol. 9115, pp. 287–308. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19488-2_15
5. Conforti, R., Dumas, M., García-Bañuelos, L., La Rosa, M.: Beyond tasks and gateways: discovering BPMN models with subprocesses, boundary events and activity markers. In: Sadiq, S., Soffer, P., Völzer, H. (eds.) BPM 2014. LNCS, vol. 8659, pp. 101–117. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10172-9_7
6. Aho, A.V., Corasick, M.J.: Efficient string matching: an aid to bibliographic search. Commun. ACM **18**(6), 333–340 (1975)
7. Karp, R.M., Rabin, M.O.: Efficient randomized pattern-matching algorithms. IBM J. Res. Dev. **31**(2), 249–260 (1987)
8. Knuth, D.E., Morris, J.H., Pratt, V.R.: Fast pattern matching in strings. SIAM J. Comput. **6**(2), 323–350 (1977)

Social Network Analysis

Health, Grades and Friendship: How Socially Constructed Characteristics Influence the Social Network Structure

Sofia Dokuka^(✉), Ekaterina Krekhovets, and Margarita Priymak

Center for Institutional Studies, NRU HSE, Moscow, Russia
{sdokuka,ekrekhovets,mvpriymak}@hse.ru

Abstract. Homophily - tendency for people to form social connections with similar others - is one of the key topics in social network analysis. It indicates to what extent people tend to be similar to their friends and in what dimensions. For the long time homophily was just an index of the social similarity, but for the recent years the interest for the homophily formation, dynamics and multidimensionality increased. In this paper we investigate the homophily in such social constructed behavior as food consumption and academic achievements. The study of body mass index in social network context reveals the presence of homophily, which means that persons with similar constitution are more likely to be interconnected with each other. Interestingly, that healthy food consumption has no impact on social network formation, but there is homophily based on fast food consumption. Thus, ‘bad habits’ are stronger forces for the social ties formation. This results show that social constructed behavior is an important component on the process of social network formation.

Keywords: Social networks · Homophily · Student networks · Health Food consumption · Academic achievements · Higher education

1 Introduction and Related Works

Social network analysis aims to identify why people form ties with each other. One of the most consistent mechanisms of social network formation is *homophily* [1]. Homophily is the tendency of people to be connected with similar others. For example, friends tend to be similar in their habits, lifestyle, values and so on. The homophily studies have a long history from the key paper by Lazarsfeld et al. [2]. The tendency of actors to be connected with similar other was found in different types of social networks such as friendship [3], social support [4], comembership [5], romantic relationships [6] and so on [1]. Usually homophily is traced by gender, race and ethnicity, but in recent papers the interest for the role of health, academic performance, online behavior, similar cultural tastes in homophily formation increased [7].

Despite the huge number of homophily-related articles in the literature, usually authors only fix the presence of social similarity of connected people.

The detailed studies of homophily formation [8], evolution [9] and multidimensionality appeared just in the last few years due to the development of statistical instruments and special network software, for example stochastic actor-oriented models and *RSiena* package [10]. The goal of this paper is to investigate the extent of homophily based on health-related behavior and academic achievements in case of economic freshmen. Both these two types of behavior are *socially constructed*. It means people tend to pay attention to the behavior of others in this particular contexts and change in order to conform the. As was shown in [11], students tend to be influenced by their peers and adjust their level of academic performance to the level of their friends. In [12] student authors show that friends and close people tend to assimilate food-related behavior of each other.

Both academic performance and health are key components of the human capital [13]. Academic performance is crucial factor of occupational and professional success [14–16]. The academic results are closely interconnected with abilities [17] and socio-economic status [18] of the person.

The role of health is much more complex. Health can impact on personal outcomes in two totally different directions, let's consider two examples in the case of health and academic performance. On the one hand, students with higher levels of health have more energy for their studies, they can even sleep less and invest this free time on their classes and homework. On the other hand, maintaining the health level needs a lot of investments, including time investments. For example, the attention to health can divert people from the academic process. Thus, the interaction between academic performance and health is non-trivial so it requires the multilateral analysis. In this paper we concentrate on food consumption and body mass index in case of health-related behavior. The quality of food is a very important factor which influences health a lot. At the same moment it is not very sensitive in terms of the survey data collection, so students can answer such questions free.

Using information about students' grades (we used grade point average as proxy for academic performance), information about health and social networks of students we show that both academic achievements, food-related behavior and body mass index impact the structure of the social networks. Students tend to form homophilous ties and segregate based on the level of their academic productivity, consumption of the fast-food and body mass index.

The structure of the paper as follows. In the second part we provide the literature review on the role of health and academic performance in the social network formation. Third part of the paper describes the data, including the data gathering processes and descriptive statistics. Fourth part introduces the results. In fifth section we conclude and discuss the further research directions.

2 Academic Achievements and Health in Social Network Formation

According to Wasserman and Faust [19] social network perspective gives an option to express the patterns and regularities of the social environment in formal

definitions. In this paper we focus on the role of academic performance and health-related behavior in homophily formation, and, broadly speaking, in social tie formation.

Academic performance is usually considered as individual characteristic which is mostly influenced by abilities, background and socio-economic status, however there is another important component of academic productivity - social environment. Academic productivity and social networks are closely bonded and tend to influence each other over time [20, 21]. There are a lot of empirical evidences which show that student tend to select their social environment (for example, choose friends and advisers) based on their levels of academic performance [20, 22]. For example, students with high grades are more likely to be popular within the environment of high achievers in case of friendship networks. The impact of social networks on academic achievements was also found in empirical settings [20–23]. It means that students tend to adjust their levels of academic performance to the level of their friends. For example, if students' friends are good in their studies she will also invest time and effort to show high academic results. The influence of high academic achievements works even in the case of online social networks which also represent the level of interpersonal communication. Vaquero and Cebrian show that students, who are successful in the university, are much more likely to be connected with other highly achieving peers and build strong connections within such online social environment [24], while low achievers are much more likely to stay on the network periphery and avoid close connections with their peers.

Health is one of the most important parts of the human capital [13], thus it may have significant impact on many spheres of life, including social environment. The impact of health on social networks formation and evolution was traced in many cases. One of the most prominent studies is one by Christakis and Fowler [12]. Using data from Framingham Heart Study authors show that obesity spreads through the social network. Despite the fact that this paper was controversially perceived in scientific community [25, 26], the role of the social networks on health-related behavior (on the weight and healthy food in particularly) became one of the most interesting and intriguing chapters in the network analysis and facilitate a lot of studies on social networks and health. In further studies researchers find mutually dependent relationship between the adolescent friendship networks and their physical activities, thus students tend to select friends based on similarity in their physical activities and adopt the levels of physical activity of their friends level [27]. In [28] authors investigate another side of health-related behavior - smoking. They show that social networks play important role in smoking spread among adolescence friendship networks.

3 Data

3.1 Description of the Case

In this study we use data on students social networks, individual characteristics of food consumption and body mass index and academic performance.

We investigate the first year students from one of the selective Russian Universities. Data were gathered in October of 2016.

Students can be matriculated in Russian universities by different trajectories. Firstly, the winners and awardies of the all-Russian and international Olympiads are accepted on tuition free places without extra exams. Secondly, students with higher exam scores are accepted on tuition free places. Thirdly, students with lower exam grades can be accepted on full tuition. The detailed description of enrollment process in Russia can be found in the following papers [9, 22].

We study the cohort of first year students of economic department who are divided into four study groups, the average size of the group is 25 persons. Lectures are usually given to the whole cohort, while seminars are delivered to the each study group separately, thus students are most actively communicate within their study groups, rather with their other peers.

Every two month students pass exam sessions. The final exam scores in the most of the subjects are cumulative and consist of both the students' performance during the semester and their final test results. In the University under investigation is the public grading system, which means that students' grades are open and publicly available. In the end of each semester university administration publishes the final rating of students based on the level of their academic results [22] on the university web-site. Top achievers receive additional financial aid from the University.

3.2 Data Collection Procedure

The data for this study were gathered from two sources: students survey and administrative database. At the beginning of the academic year we asked students about their friendship social networks (*Please, name your course mates with whom you spent most of your time*) and health-related behavior (healthy-food consumption, fast food consumption, eight and weight). Based on information of height and weight we calculated the body mass index. Questionnaires were handed out at the end of lectures and students had approximately twenty minutes to fill them out. Students were encouraged not to discuss questions with their peers.

To identify the patterns of food consumption we used three indicators: the information about healthy food consumption, the information about fast food consumption and body mass index. The question about healthy-food consumption was as the following '*Estimate the quality of your food from 1 to 7. 1 is the junk food and 7 is the healthy food*'. The attitude toward fast-food we measured using the following question '*How often do you consume fast-food?*' with possible options: practically every day, 3–4 times a week, 1–2 times a week, very seldom.

Body mass index is defined as a person's weight in kilograms divided by the square of her height in meters [29]. According to World Health Organization (WHO) BMI for healthy person should vary between 18.5 and 25. BMI under 18.5 means underweight, over 25 - overweight. If persons' BMI is higher than 30, it means obesity. We asked students about their weight and height and based on self-reports calculated their BMI.

Information about students' exam scores and group affiliation were gathered from administrative database.

3.3 Descriptive Statistics

In this study we investigated the cohort of first year students from economic department of one of the Russian universities. Data were gathered at the beginning of the academic year, in October 2016. We analyzed data about 92 students (about 90% of the whole cohort). The sample is gender biased, there are 21% of male and 79% female in the cohort. The descriptive statistic for the social networks and individual students characteristics is presented in Tables 1 and 2.

Table 1. Friendship social network descriptive statistics

Parameter	Value
Number of nodes	92
Number of edges	263
Density	0.03
Reciprocity	0.38
Transitivity	0.34
Degree assortativity	0.07

The structure of student friendship network is typical for friendship social networks [10]. It is reciprocal and transitive, which means that students tend to form mutual friendship ties [30] and befriend the friends of their friends [31]. Low density shows that students are selective in their friendship nominations. Low degree assortativity means that nodes seek to connect with higher degree nodes [32]. The social network under investigation was collected during the very beginning of the school year. In the mid of October students befriend each other, but they do not have enough time and resources to become real friends and pass together difficulties (such as exam session) (Fig. 1).

Table 2. Individual characteristics

Parameter	Mean	Minimun	Maximum
Academic performance	5.5	0	10
Healthy food consumption	5.5	2	7
Fast food consumption	3.4	1	4
Body mass index	20.3	15.3	31.4

Statistics for the academic performance and health-related behavior shows that students mostly prefer eat healthy food and avoid fast-food. The body

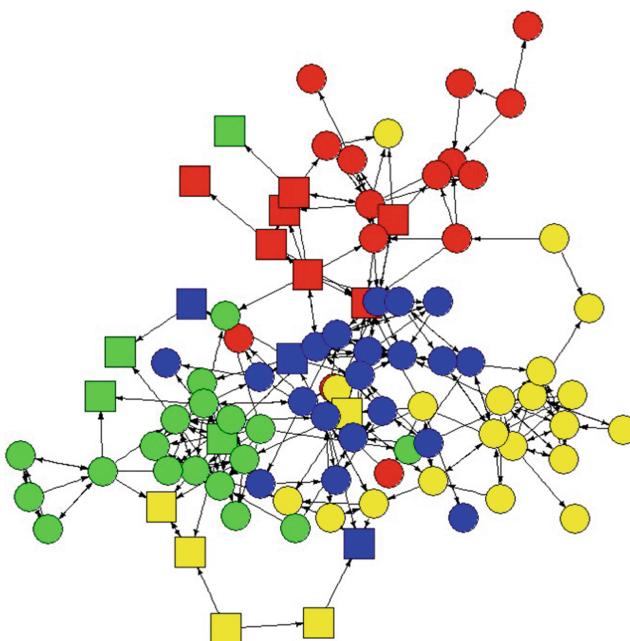


Fig. 1. The student friendship network. Nodes are students, edges are friendship connections between them. Males are squares, females are circles. Nodes of the same color are students from the same study groups. (Color figure online)

mass index varies from very low (15) to very high (31) which means that there are few thick and obese persons in our sample. The mean of BMI is about 20, which means that students are mostly thin. Mostly it consists of young active girls, with defined beauty standards. Such results can be biased because of the specific gender balance. Due to the structure of the social network our sample is quite homogenous. It heavily impacts the structure of the BMI distribution and makes it positively skewed, which shows the dominance of thin persons in the cohort.

Table 3. Individual characteristics

GPA/BMI	Below 18	18–25	Above 25
Low achievers	1	19	2
Below-average	5	20	0
Above-average	5	23	1
High achievers	3	13	0

Cross-tabs shows the dominance of persons with normal body mass index. Still, there are few persons with low BMI (thin) and high BMI (obesity).

4 Method

We used exponential random graph models (ERGM) in order to identify local network structures, which can be the result of endogenous and exogenous processes [33]. ERGM is a very widespread social network approach which allows to reveal the character structural network properties and motifs on static network snapshot.

According to Robins et al., real-world social networks can be viewed as one realization from a set of possible networks of the same size. To reveal to what extent the observed social network differs from the networks that occur by chance. The networks are usually compared by the set of characteristics which called effects. We can speak both about endogenous network-related and exogenous characteristics. Endogenous or structural effects are the functions of the network itself, without taking into the consideration the characteristics or behavior of actors. For example, reciprocity means the formation of mutual ties between two persons, transitivity describes the friendship between three persons. Exogenous effects based on the individual characteristics of actors or their dyadic covariates. For example, popularity of males within the social network is the individual exogenous effect, while the gender-based homophily and formation the friendship ties between males is the dyadic covariate.

The important part of exponential random graph models is the partial conditional independence, which means that two potential social network ties are considered to be partial conditionally dependent if they (i) share a common actor, or (ii) if they are the parts of four-cycle [34]. This is a crucial assumption for the social network data, because it takes into the consideration the complex and interconnected nature of social ties.

The notation of ERG models as follows [33].

$$Pr(Y = y) = \frac{1}{k} \exp(\sum_A (\eta_A g_A(Y)))$$

where η_A is the parameter corresponding to A structural configuration, $g_A(Y)$ is the network statistics, corresponding to A configuration and k is the normalizing quantity which ensures that the function has a proper probability distribution. This function describes the probability of tie formation.

Exponential random graph models are widely used in social network studies in different substantial areas, e.g. education, health [35], management [36], etc. These network models are implemented in *statnet* and *ergm* packages in R statistical environment [37].

It is important to underline that exponential random graph models can describe the network structures and motifs, but due to the static nature of network data the mechanisms of network processes can not be revealed in such setup. Thus we can not claim about the causal links, but the presence of correlations still shed lights on the nature of interconnections between social networks and individual characteristics.

5 Results

The results of the exponential random graph modeling are in Table 3.

Table 4. ERG model results

Parameter	Estimate	St. Error
Edges	-4.37***	0.67
Reciprocity	1.46***	0.25
Popularity	-0.22***	0.03
Transitivity	1.46***	0.13
Female gender homophily	0.49***	0.14
Male gender homophily	0.76***	0.21
Study in the same group	1.53***	0.15
Students with high GPA popularity	0.45+	0.27
Absolute difference in GPA	-1.35***	0.32
Students with high healthy food consumption popularity	-0.08	0.05
Absolute difference in healthy food consumption	-0.02	0.06
Students with low fastfood consumption popularity	-0.056	0.08
Absolute difference in fastfood consumption	-0.23**	0.07
Students with high BMI popularity	0.05+	0.02
Absolute difference in BMI	-0.06*	0.03

AIC: 1614 BIC: 1719

*** p < 0.001, ** p < 0.01, ** p < 0.05, + p < 0.1

Results demonstrate that academic performance is an important factor of social network formation. The negative absolute difference in GPA means that students tend to be connected with peers with similar levels of academic performance, in other words, academic achievement is the basis of homophily ties formation. The positive popularity of students with high GPA shows that academically productive persons are more likely to receive more nominations. The sum of these effects is often fixed in academic environments [20, 24] and it usually means the presence of academic performance-based homophily (Table 4).

The impact of health-related characteristics on the social network is more complex. We do not fix the presence of homophily based on healthy food consumption, but find selection in case of fast food consumption. It means that students are befriend people who have similar fast food consumption. The body mass index reveals to be an important factor of friendship, students tend to befriend peers with similar constitution. At the same time actors with high BMI turn out to be more popular within friendship network.

Modeling results show the presence of gender-based homophily, which is very widespread in social networks. Males tend to befriend male, while females are

more likely to be connected with females. Students also prefer interact with their group mates, it is essential, because, as was describes above, students spent most of their time within the study groups. The negative and significant edges effect shows that students do not tend to form social ties, and if they form connections they are involved in more complex social structures, reciprocal and transitive.

5.1 Conclusion and Future Research Directions

The complex structure of homophily is one of the most prominent subjects in contemporary social network area. In this paper we focus on the homophily formation based on different socially constructed characteristics, such as academic performance and health-related behavior. Using survey data we show that students tend to segregate based on the level of academic performance. In other words students with high academic performance form close and dense network group and avoid of interactions with their low-achieving peers. Such structure, as was previously outlined by [24], shows the absence of knowledge transfer across the classroom and can further result in decreasing of the motivation among students and even dropouts.

The role of health-related behavior in social network formation was studied in cases of BMI and food consumption. We reveal that students tend to select their friends with similar levels of fast food consumption, while healthy food consumption is not significant predictor for social ties formation. This results may be due to the nature of healthy and fast food, where the fast food is much more social rather than healthy. Body mass index also plays important role in network formation, students befriend peers with the similar levels of BMI and, at the same time, persons with high BMI are more popular within the friendship network.

Obtained results shed lights on the complex interaction between social ties and socially constructed individual characteristics, but due to the research design we can not claim on causal mechanism. Thus, to our mind the essential part of this study is the collection on the second and third longitudinal data sets and further analysis of the consequent network and behavior characteristics. As was shown in previous studies we can find very complicated mechanisms of network selection and influence in cases of both food consumption and academic performance [20, 28].

It can be also of a great interest to concentrate on more specific (or wide) characteristics of the health-related behavior. In present study we considered only food-consumption variables and body-mass index to define and access health but in the future there will be significant to improve the understanding of the variables that can be used as the proxy for the definition of health.

Acknowledgements. We would like to thank Maria Yudkevich for help and discussion. The financial support of the 5-100 Government Program and Basic Research Program at the National Research University Higher School of Economics (HSE) is greatly appreciated.

References

1. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: homophily in social networks. *Ann. Rev. Sociol.* **27**, 415–444 (2001)
2. Lazarsfeld, P.F., Merton, R.K., et al.: Friendship as a social process: a substantive and methodological analysis. *Freedom Control Mod. Soc.* **18**, 18–66 (1954)
3. Kandel, D.B.: Homophily, selection, and socialization in adolescent friendships. *Am. J. Sociol.* **84**, 427–436 (1978)
4. Suitor, J., Keeton, S.: Once a friend, always a friend? Effects of homophily on women's support networks across a decade. *Soc. Netw.* **19**, 51–62 (1997)
5. Newman, L., Dale, A.: Homophily and agency: creating effective sustainable development networks. *Environ. Dev. Sustain.* **9**, 79–90 (2007)
6. Furman, W., Simon, V.A.: Homophily in adolescent romantic relationships. In: *Understanding Peer Influence in Children and Adolescents*, pp. 203–224 (2008)
7. Lewis, K., Kaufman, J., Gonzalez, M., Wimmer, A., Christakis, N.: Tastes, ties, and time: A new social network dataset using facebook.com. *Soc. Netw.* **30**, 330–342 (2008)
8. Steglich, C., Snijders, T.A., Pearson, M.: Dynamic networks and behavior: separating selection from influence. *Sociol. Method.* **40**, 329–393 (2010)
9. Dokuka, S., Valeeva, D., Yudkevich, M.: Homophily evolution in online networks: Who is a good friend and when? In: Ignatov, D.I., Khachay, M.Y., Labunets, V.G., Loukachevitch, N., Nikolenko, S.I., Panchenko, A., Savchenko, A.V., Vorontsov, K. (eds.) *AIST 2016. CCIS*, vol. 661, pp. 91–99. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-52920-2_9
10. Snijders, T.A., Van de Bunt, G.G., Steglich, C.E.: Introduction to stochastic actor-based models for network dynamics. *Soc. Netw.* **32**, 44–60 (2010)
11. Dokuka, S., Valeeva, D., Yudkevich, M.: The Diffusion of Academic Achievements: Social Selection and Influence in Student Networks (2015)
12. Christakis, N.A., Fowler, J.H.: The spread of obesity in a large social network over 32 years. *N. Engl. J. Med.* **357**, 370–379 (2007)
13. Becker, G.S.: Health as human capital: synthesis and extensions. *Oxford Econ. Pap.* **59**, 379–410 (2007)
14. Freier, R., Schumann, M., Siedler, T.: The earnings returns to graduating with honors—evidence from law graduates. *Labour Econ.* **34**, 39–50 (2015)
15. Gleason, P.M.: College student employment, academic progress, and postcollege labor market success. *J. Student Financ. Aid* **23**, 5–14 (1993)
16. Jones, E.B., Jackson, J.D.: College grades and labor market rewards. *J. Hum. Resour.* **25**, 253–266 (1990)
17. Rourke, B.P., Finlayson, M.A.J.: Neuropsychological significance of variations in patterns of academic performance: Verbal and visual-spatial abilities. *J. Abnorm. Child Psychol.* **6**, 121–133 (1978)
18. Sirin, S.R.: Socioeconomic status and academic achievement: a meta-analytic review of research. *Rev. Educ. Res.* **75**, 417–453 (2005)
19. Wasserman, S., Faust, K.: *Social Network Analysis: Methods and Applications*, vol. 8. Cambridge University Press, Cambridge (1994)
20. Lomi, A., Snijders, T.A., Steglich, C.E., Torló, V.J.: Why are some more peer than others? Evidence from a longitudinal study of social networks and individual academic performance. *Soc. Sci. Res.* **40**, 1506–1520 (2011)
21. Flashman, J.: Academic achievement and its impact on friend dynamics. *Sociol. Educ.* **85**, 61–80 (2012)

22. Dokuka, S., Valeeva, D., Yudkevich, M.: Formation and evolution mechanisms in online network of students: the Vkontakte case. In: Khachay, M.Y., Konstantinova, N., Panchenko, A., Ignatov, D.I., Labunets, V.G. (eds.) AIST 2015. CCIS, vol. 542, pp. 263–274. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26123-2_26
23. Sacerdote, B.: Peer effects with random assignment: Results for Dartmouth roommates. *Q. J. Econ.* **116**, 681–704 (2001)
24. Vaquero, L.M., Cebrian, M.: The rich club phenomenon in the classroom. *Sci. Rep.* **3** (2013)
25. Cohen-Cole, E., Fletcher, J.M.: Is obesity contagious? Social networks vs. environmental factors in the obesity epidemic. *J. Health Econ.* **27**, 1382–1387 (2008)
26. Fowler, J.H., Christakis, N.A.: Estimating peer effects on health in social networks: a response to Cohen-cole and Fletcher; Trogdon, Nonnemaker, Pais. *J. Health Econ.* **27**, 1400 (2008)
27. De La Haye, K., Robins, G., Mohr, P., Wilson, C.: How physical activity shapes, and is shaped by, adolescent friendships. *Soc. Sci. Med.* **73**, 719–728 (2011)
28. Mercken, L., Snijders, T.A., Steglich, C., Vartiainen, E., De Vries, H.: Dynamics of adolescent friendship networks and smoking behavior. *Soc. Netw.* **32**, 72–81 (2010)
29. Quetelet, A.: *Physique sociale, ou essai sur le développement des facultés de l'homme*, vol. 2. C. Muquardt (1869)
30. Gouldner, A.W.: The norm of reciprocity: a preliminary statement. *Am. Sociol. Rev.* **25**, 161–178 (1960)
31. Watts, D.J., Strogatz, S.H.: Collective dynamics of ‘small-world’ networks. *Nature* **393**, 440–442 (1998)
32. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* **286**, 509–512 (1999)
33. Robins, G., Pattison, P., Kalish, Y., Lusher, D.: An introduction to exponential random graph (p^*) models for social networks. *Soc. Netw.* **29**, 173–191 (2007)
34. Prell, C.: *Social Network Analysis: History, Theory and Methodology*. Sage, Thousand Oaks (2012)
35. Harris, J.K., Carothers, B.J., Wald, L.M., Shelton, S.C., Leischow, S.J.: Interpersonal influence among public health leaders in the united states department of health and human services. *J. Public Health Res.* **1**, 67 (2012)
36. Ellwardt, L., Labianca, G.J., Wittek, R.: Who are the objects of positive and negative gossip at work?: A social network perspective on workplace gossip. *Soc. Netw.* **34**, 193–205 (2012)
37. Team, R.C.: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, 2013 (2014)

Dynamic Semantic Network Analysis of Unstructured Text Corpora

Alexander Kharlamov^{1,2,3}(✉) , Galina Gradoselskaya³ , and Sofia Dokuka³ 

¹ Institute of Higher Nervous Activity and Neurophysiology of RAS, Moscow, Russia
kharlamov@analyst.ru

² Moscow State Linguistic University, Moscow, Russia

³ Higher School of Economics, National Research University, Moscow, Russia

Abstract. The natural language structure can be viewed as weighted semantic network. Such representation gives an option to investigate the text corpus as the model of the subject domain. In this paper we propose the mechanism of the semantic network identification and construction. We apply the methodological instrument for the social media text analysis and trace the dynamics of the discussions about 1917 year within the internet communities. Network changes illustrate the changes of the interest to different topics. The proposed mechanism can be used for the monitoring of the different social processes and phenomenal in online social networks and media.

Keywords: Text mining · Unstructured text processing
Neural networks · N-gram text model · Semantic networks
Thematic trees · Structural analysis · 1917

1 Introduction

The analysis of unstructured texts (for example, text from social media) is one of the key directions in text mining research. Text analysis of big data sets often conducted by unigram model [1].

Models based on the latent semantic analysis, probabilistic latent semantic analysis and latent Dirichlet allocation can detect keywords - topics - hidden in the structure of the text and show the relationship between text units (sentences, paragraphs and whole texts in the corpus) and words found in them by identifying relationships of words and themes and comparing these with the text units.

To identify the structural relationships between text units and words composing them, the latent semantic analysis (LSA) method is used [2]. The latent semantic analysis is based on linear algebra and presents a method of reducing the matrix dimension by means of matrix decomposition. It uses a vector representation of text units such as “bag of words”. Thus, a text or a corpus of texts as a set of text units (sentences - d) is represented as a numerical matrix where rows correspond to words (w) included in the text (text corpus), and columns - to text units (sentences - d). Introduction of so-called hidden themes (z) using a

diagonal matrix $\widetilde{\Sigma}$ where diagonal elements correspond to weights of the themes; grouping words of the text with these themes using a matrix \widetilde{U} to display the word space in the theme space; and presentation of text units (sentences) in the space of these themes by using a matrix \widetilde{V} to present text units in the space of these themes allow the product of these matrices $P = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^T$ to be decomposed, which results in identification of hidden themes (weight characteristics of the diagonal elements of the diagonal theme matrix $\widetilde{\Sigma}$). The number of themes is specified in advance.

The Probabilistic mathematical model (Probabilistic Latent Semantic Analysis, pLSA) is similar to the previous model class based on latent semantic analysis [3]. The difference between them is in the process of model building. Compared with the conventional latent semantic analysis, the probabilistic latent semantic analysis is based on the assumption that the said correspondences (words of the text and themes, themes and text units) are described with probabilities of their occurrence.

To determine parameters of the model with a pre-specified number of themes k (as in LSA), an EM (Expectation Maximization) algorithm is used, that is an iterative procedure for calculating hidden variables by maximizing the likelihood function. Then the thematic model may be written in a matrix form, as in the latent semantic analysis: $P = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^T$.

Latent Dirichlet allocation (LDA) is a further development of the probabilistic latent semantic analysis [4]. In this approach, the same terminology is used, but an additional language model θ is introduced (or at least a text corpus - the domain model), which is defined by a family of continuous multi-dimensional probability distributions of non-negative real numbers $\text{Dir}(\alpha)$ parameterized by the vector α , where matrix β of size $k * V$ is introduced, which is external to the Dirichlet allocation, where columns correspond to themes of the text (text corpus) fixed, as in the two previous models by the number (k), and rows - to words of the language model dictionary (or at least the domain model dictionary). Evaluation of α and β parameters of the model is also based on the EM-algorithm, but, unlike the above two models, this evaluation cannot be performed analytically, but only through a variational EM-procedure.

But Latent Dirichlet allocation has a major drawback, the lack of convincing linguistic substantiation. The assumption that all allocations $\theta_d, d \in D$ where D is a set of text sentences, are generated by the Dirichlet allocation (and by the same one) seems rather arbitrary. The same can be said about generation of the set of α_t allocations for all themes $t \in T$, where T is a set of themes.

All three types of thematic models are considered exclusively within a monogram model [1]. The monogram model in these approaches is used exclusively because of great complexity of thematic models that increases even more when using the Dirichlet allocation.

The use of n-gram text model gives an option to identify not only single topics, but the thematic trees, which show the interplay between different topics.

The application of this method to the sequence of texts in dynamics can show the mechanisms and evolution of the social processes.

The paper consists of two major parts. In the first part we present the theoretical background and algorithms of the text analysis in case of n-gram models. Second part is an example of structural text analysis on the corpus of text from social media.

2 Structural Text Analysis on the Base of N-gram Text Model

The structural analysis of the text is the identification of the key text elements and connections between them. Thus, the resulting semantic network consists of the set of nodes (words and sustainable expressions) and edges between them (which means the joint occurrence in compact text structures, e.g. sentences). Such network can be used for the identification of the connections between the text concepts.

2.1 Structural Text Representation

To form a homogeneous semantic (associative) network, a frequency portrait of the text is created. It contains information on the frequency of occurrence of key concepts of the text, represented as the root bases of the corresponding words, or their stable combinations occurring in the text, as well as their joint (pairwise) occurrence in semantic text fragments (for example, in sentences) [5].

Root bases, instead of words, are used to exclude the influence of language inflectivity on the quality of the analysis, for which the morphological processing of the text is carried out using a pre-prepared morphological dictionary (the dictionary of inflected morphemes) - a first-level dictionary - $\{B_i\}_1$. As a result, a second level dictionary is formed - $\{B_i\}_2$ - dictionary of root words bases (and stable combinations of root bases). Dictionary of the syntactic level - a dictionary of syntax B_{i3} is not used in this algorithm. But the fourth level it is formed $\{B_i\}_4$ - a dictionary of semantic level - couples of words.

At this stage frequencies are detected p_i the occurrence of root B_{i2} key concepts (obtained as a result of morphological analysis) and their stable combinations, and frequencies p_{ij} their pairwise occurrence in sentences of the text B_{i4} .

2.2 Homogeneous Semantic Network

The result is a primary (frequency) associative network N as a set of asymmetric pairs of concepts $\langle c_i c_j \rangle$, where c_i and c_j - are text concepts connected with each other by association relationships (joint occurrence in some text fragment) [5]:

$$N \cong \langle c_i c_j \rangle > \quad (1)$$

Otherwise, the semantic network can be represented as a set of stars $\langle c_i \langle c_j \rangle \rangle$, where $\langle c_j \rangle$ - a bunch of $\langle c_j \rangle$ closest associates of the key concept c_i :

$$N \cong \{z_i\} = \{\langle c_i \langle c_j \rangle \rangle\} \quad (2)$$

Associates c_j of the main concept c_i are its semantic features and allow one to interpret it meaningfully.

2.3 N-gram Text Model. Ranking of Concepts

The use of the n-gram model of the text in topic modeling makes it possible to correctly interpret the results from a linguistic point of view. In order to perform this, we introduce a different understanding of the text topics, compared with those used in approaches based on latent semantic analysis [5].

In the bigram model, we will consider those “second” (in the bigram: “first word - second word”) the words of the largest rank in the text that are related to the largest number of “first” words. In the trigram model, we will consider those “third” words (second “second”) of the highest rank, which are related to the largest number of “second” words that have the greatest rank in terms of the bigram model. And so on to the n-th order of the model. Then in the n-gram model, we will consider those “n-th” words of the greatest rank that are related to the largest number of “(n-1)-th” words of the (n-1)-gram model. Thus, the thematic trees are automatically formed, in which the main topics of the text are the n-th level (n-th words) of the highest rank, their sub-themes are the (n-1)-th level ((n-1)-th words), their sub-sub-themes - the theme (n-2)-th level, etc.

We start by considering the themes of the first level (bigram model of the text). If we consider the emergence of a sequence of two words in the text, we get a bigram model. For every second word w_j strings of two words (w_i, w_j) the first word of the string w_i (“from left to right”) is the theme: $w_i \cong t_i^2$ (index 2 - because we consider bigram model). Combine all pairs of words with identical topics in stars. In this case, the node of the network corresponding to the word $w_i \cong t_i^2$, is the root vertex of one of the thematic trees (in this case, stars). Since the probability of the appearance of a string of two words (right-side model) in the text $p(w_i, w_j) = p(w_j|w_i)p(w_i)$, probability of appearance of a topic $w_i \cong t_i^2$ in the bigram model there is a sum of the probabilities of the appearance of pairs with the same first word (probability of an star):

$$p(t_i) = p(w_i^2) = \sum_{j=1}^{J_i} p(w_i, w_j) = \sum_{j=1}^{J_i} p(w_j|w_i)p(w_i^1),$$

where J_i - number of words w_j (associates of w_i star), connected with the first word w_i . Probabilities $p(w_i^1)$ - are the initial probabilities of word distribution in the text. We introduce the conditional concept of the “theme” $p(w_i^1) \cong t_i^1$ for the monogram distribution. And so for every second word w_j the first word of w_i (“from left to right”) is the theme: $w_i \cong t_i^2$:

$$p(t_i^2) = p(w_i^2) = \sum_{j=1}^{J_i} p(w_i, w_j) = \sum_{j=1}^{J_i} p(w_j|w_i)p(t_i^1), \quad (3)$$

In order for the total sum of probabilities $P(t_i^2)$ to be equal to one: $\sum_{n=1}^N P(t_n^2) = 1$, where n - is the number of topics, it is necessary to normalize the t_n^1 sums. In general, the number of topics coincides with the number of all words in the text $T = W$, but usually select only a few main themes: $T \leq W$. Rationing is carried out on the sum of all topics t_i^1 :

$$p(t_i^2) = \frac{\sum_{j=1}^{J_i} p(w_j|w_i)p(t_i^1)}{\sum_{i=1}^T \sum_{j=1}^{J_n} p(w_j|w_i)p(t_i^1)} \quad (4)$$

Here $p(t_i^1)$ in the Formula (3) means the probability of occurrence of a single word (that is, the probability from the monogram model). And $p(t_i^2)$ - is the probability of occurrence of an star from the bigram model. Moreover, the main word of an star in terms of the bigram model is a topic for its closest associates - semantic features - “second” words. For a sequence of three words in length, that is, a trigram model of the text:

$$p(w_i w_j w_k) = p(w_k|w_i w_j)p(w_i w_j) = p(w_k|w_i w_j)p(w_j|w_i)p(w_i).$$

Then the probability of the appearance of a line from the first two words in a string of three words can be obtained, as in (3) by summing over the third word:

$$p(w_i w_j) = \sum_{k=1}^{K_j} p(w_i w_j w_k),$$

and the probability of the appearance of the first word in a string of three words can be obtained by summing over the second word:

$$p(w_i) = \sum_{j=1}^{J_i} \sum_{k=1}^{K_j} p(w_i w_j w_k).$$

The probability $p(t_i^3)$ is calculated in accordance with the expression (3) in the bigram model, $p(w_i) \cong p(t_i^2)$, and the probability of the second word of the pair appearing in the text, depending on the appearance of the first word of the pair in the text, is the same, regardless of the grammability of the model text, as follows from the network representation of the text (interrelated pairs of words remain the same):

$$p(t_i^3) = \frac{\sum_{j=1}^{J_i} p(w_j|t_i^2)p(t_i^2)}{\sum_{i=1}^T \sum_{j=1}^{J_i} p(w_j|t_i^2)p(t_i^2)} \quad (5)$$

where t_i^2 and t_i^3 mean, respectively, the theme in accordance with the bigram and trigram models, and the number of topics, as before, can correspond to the number of words in the text $T = W$, but is usually limited to a strong-willed

solution up to $T \leq W$. To the root nodes t_i^3 , through the “second” node t_i^2 of the star z_i^3 , the stars of the bigram model z_i^2 are attached:

$$z_i^2 = \{< t_i^2 \{t_i^1\} >\}.$$

The same procedure applied for the analysis of four-gram and more grammar models will lead to the following sets of root nodes, to which all the graphs (thematic structures) obtained at the previous stages of analysis are attached. That is, we can calculate the weights of the thematic nodes of the semantic network of text, taking into account their dependence in a sequence of four, and ultimately of n words.

In other words, we have an iterative re-weighed procedure that allows us to find the probability of occurrence of those t_i^n (in the case of an n -gram model) in the text:

$$p(t_i^n) = \frac{\sum_{j=1}^{J_i} p(w_j|w_i)p(t_i^{n-1})}{\sum_{i=1}^I \sum_{j=1}^{J_i} p(w_j|w_i)p(t_i^{n-1})} \quad (6)$$

where $p(t_i^1) = p(w_i)$ and $p(w_j|w_i)$ is the probability of occurrence of the next word of the text for all the iteration steps, provided the previous word appears.

2.4 Thematic Structure of the Text

Then the star, in which the main theme is the theme of the n -gram model t_i^n , and the secondary words (the closest associates) are the themes of the $(n-1)$ -gram model t_i^{n-1} , to which the stars are attached, in which the main words are the corresponding topics of the t_i^{n-1} $(n-1)$ -gram model, and the secondary topics are the topics of the t_i^{n-2} $(n-2)$ -gram model, to which, in turn, are the same. The corresponding stars of lower-level models are attached, called the thematic tree.

For an accurate assessment of the semantic weights of concepts, we use in this way the weights of all concepts associated with them, i.e. the weight of the whole semantic condensation. As a result of the iterative reranking procedure, the most important concepts are those related to the largest number of other concepts with a large weight, that is, those concepts that pull together the semantic structure of the text. The semantic weights of key concepts obtained in this way show the significance of these concepts in the text.

3 Semantic Network Dynamics

So, we got a structure that consists of weighted concepts, and describes the relationship between these concepts in the text (the corpus of texts). We can take the concepts of the highest (above a given threshold) rank, which we call key concepts of the text. We will create such networks for different corpus of texts that arise historically at different time moments, as describing the dynamics of a certain process.

Consider a sequence of like-named stars, belonging to different temporal semantic sections - semantic networks, and call it an elementary process π [6]:

$$\pi = z_i(t_i) \Rightarrow z_i(t_2) \Rightarrow z_i(t_3) \Rightarrow \dots \quad (7)$$

where $z_i(t_k)$ is the specific star at the time t_k . The weight of the key concept at a given time, which determines its rank in the semantic network $w_i(t_k)$. We consider the aggregates of such chains of stars entering into semantic networks, obtained as a result of the analysis of the corpus of texts at different time intervals.

A semantic network built on a text written later and describing the same structure may differ from the first, because it represents the text that is relevant to the state of the described process at a later time than the previous one. A network may contain the same key concepts, but may not contain some of them that have dropped out of the described structure, and may include other concepts that appeared in the text described by the text during this time. And, most importantly, the weight characteristics of the concepts contained in the network may differ from their weight characteristics, which characterize these concepts in the first network.

We connect the same key concepts of both networks with links, the thickness of which will be proportional to the weight of the key concept. If the concepts in both networks have the same weight, the connection has the same thickness from the network to the network. If the concepts have different weights, the connection connecting them either thickens or thinens, demonstrating the dynamics of the states of key concepts, and, thus, the dynamics of the network states as a whole.

If we take the texts of the next time slice, and build another network, and attach it to the two previous ones, we will have a picture of the unfolding of the structure of the process texts described in the text corpus in time. Such a model of the dynamics of the process is obvious, convenient for research (the network as a static semantic slice of the structure under study is convenient for navigating through it because of the associativity of the links between the key concepts), and possesses numerical characteristics, which makes it convenient for analytical study of processes, and as a consequence, convenient for automatic analysis.

Finally, in order to investigate a particular process in its dynamics, we choose the key concepts of the semantic network, which are lexical and psycholinguistic marks of this process, and we will explore the dynamics of the development of quantitative characteristics of these concepts.

We remove all concepts from all networks, except for the mentioned labels. In this case, the remainder of the text dynamics model becomes a model of the dynamics of the process under study. Moreover, the numerical characteristics of the remaining key concepts of the network characterize the state of the process at the current time, and their change, from the time slice to the time slice, characterizes the dynamics of the process in time.

For all lexical marks characterizing the process, the product of the label status (on the “good-bad” scale) S_i is calculated for its rank. The products \prod_i obtained for each label M_i are summarized by all labels: Thus, the total characteristic $\prod(l)$ of the time section l of the evaluated process is obtained.

4 Empirical Application: Social Media Text Mining

The approach to the analysis of the dynamics of processes based on the processing of poorly structured texts can be clearly demonstrated by the example of evaluating the tonality of texts taken from an arbitrary domain. A sample of texts dedicated to the Great October Socialist Revolution was compiled. To this body texts were used methods of thematic analysis, as well as methods of automatic semantic analysis, the algorithms of which are described above. The concepts of a homogeneous semantic network obtained on the basis of this corpus of texts were ranked and their ranks were used to evaluate the tonality by selecting lexical labels characterizing texts from both positive and negative connotations. The dynamics of the tonality change in several time intervals was analyzed. Some key concepts were also analyzed on the basis of an analysis of their closest associates as their semantic features.

The original corpus of texts (its volume was 20 MB in text format) was formed by the usual search by keywords. Main ones are: 1917, October, October uprising, October revolution, Great October socialist revolution, October revolution, socialist revolution. Auxiliary: Aurora shot, Winter storm, World War I, civil war, Lenin, Trotsky, Kerensky, Kornilov, RSDLP, Iskra, Entente, intervention, Bolshevik, Menshevik, Menshevism, Reds, White.

Collection of a text file was carried out on the most popular social networks: Facebook, Vkontakte, LiveJournal. In the array were both posts, and comments to them, if they contained one or more keywords. The message collection period is from June to September 2015.

As the main tool for statistical analysis of texts, the TextAnalyst software system was developed by MICROSYSTEMS, Ltd., Moscow, Russia. The system for processing textual information is based on the use of structural properties of the language and text that can be detected using statistical analysis, implemented on the basis of hierarchical structures from dynamic associative storage devices - neural network structures based on neuron-like elements with time summation of signals [5].

The kernel of the system implements the following functions. Normalization of grammatical forms of words. Automatic selection of basic concepts of text (words and stable word combinations) and their interrelations in the text, with calculation of their relative importance. Formation of the representation of the semantics of the text (the set of texts) in the form of a homogeneous semantic network. The core of the system has the following structure (see Fig. 1).

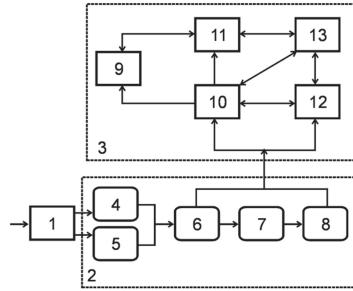


Fig. 1. The system for the semantic analysis of texts contains a primary processing unit (1), a linguistic, and a semantic processor. The linguistic processor (2) consists of dictionaries: (4) delimiter words, (5) auxiliary words, (6) common words, and (7) inflectional and (8) root morphemes. Semantically, the processor (3) in turn contains: (9) a block of references to the text, (10) a semantic network generation unit, (11) a semantic network storage unit, (12) a concept highlighting unit, and (13) a control unit

4.1 Topic Modeling

The thematic analysis of the body of texts is carried out using the n-gram textual model as the identification of a minimal tree-like subgraph of the semantic network with the largest weight of the root node. The thematic analysis shows that, on the one hand, the body of texts turned out to be very colorful, and, on the other hand, simultaneously with the revolution, several topics are considered: war, state structure, national peculiarities, and all this with reference to the main theme “Russia”.

4.2 Dynamics of Semantic Network

The corpus of the texts was divided into four parts according to the time of the corresponding texts. For each section of the corps, a semantic network was formed with the ranking of concepts. Further, for the sociologist's chosen key words of the subject domain “VOSR” with positive and negative connotation, their weight characteristics for the corresponding slices of the hull were found (see Tables 1 and 2).

Keywords (in fact the assembled array - the most massive). Positive, neutral: Lenin, Stalin, the great Russian revolution, Russia, the Soviet Union, Trotsky, Nicholas II, the Great October Revolution, the proletariat, the bourgeoisie, the communist, communism, red, white, intervention, whiteguard, soviet, historical, war, peace, world, leader, 1917, state, Russian, class, struggle, world (world revolution). Negative: coup, (October coup), Jewish, Jewish revolution, Ukraine, Makhno, Hitler, Germany.

Table 1. Weight characteristics of some lexical labels with positive connotations.

Key words	June	July	August	September
Positive				
White armies	39	5	16	0
Struggle	8	99	100	0
Bourgeoisie	99	46	99	99
Vladimir Iluich	82	12	31	91
Leader of world proletariat	13	6	99	0
Josif Stalin	97	5	9	8
Class struggle	98	5	45	10
Red army	99	97	99	96
October revolution	99	65	12	89
Nicolai II	0	26	99	95
Proletariat	99	95	99	98
Russian	0	53	100	0
USSR	100	99	99	100
Soviet regime	28	99	99	99
Normalized	46	56	46	37

Table 2. Weight characteristics of some lexical labels with negative connotations.

Key words	June	July	August	September
Negative				
Hitlerite Germany	94	99	13	0
Hitler	99	25	99	99
Jewish	0	99	0	0
Hebrew	5	3	54	0
Makhno	0	0	5	99
October regime change	0	28	0	0
Regime change	68	98	99	99
Normalized	46	56	46	37

Summarizing all the keywords, after normalizing for their number, the following characteristics of the change in the ratio of authors to this subject domain were obtained (see Table 3). It should be noted that the example of the domain was chosen randomly, that is, the dynamics of changing the opinions of authors on a given topic within such a short time is exceptionally random.

Table 3. Averaged summary characteristics of lexical labels with positive and negative connotations

Key words	June	July	August	September
Positive	29	36	53	49
Negative	46	56	46	37

4.3 Contextual Environment of the Most Significant Concepts

Significant interest for meaningful analysis is the contextual environment of the most significant concepts. The significance of the concepts was determined by the size of the weight coefficients and their stability during the four months of observation.

At once it can be noted that it was difficult to single out positive and negative concepts according to the data of the massif about 1917, as the current specificity of the consciousness of visitors to social networks affected. There were very few independent original judgments, texts on the one hand were fact-based, on the other hand, enumeration of cliches. The historical figures were actively discussed, names were enumerated and all stereotypes of perception associated with them were listed, from pre-revolutionary to post-perestroika. This, perhaps, negatively characterizes the audience of social networks, but on the other hand it simplifies the formalized analysis of the semantic structures that are produced by the mass consciousness of the audience of social networks.

The most stable, with high weights were the following concepts. Conditionally positive, neutral: Vladimir Ilyich, Ilyich, Leader of the world proletariat, Joseph Stalin, Nicholas II, Trotsky, Red Army, Belogvardeysky, Intervention, Bourgeoisie, Proletariat, Soviet power, October Revolution, Socialist Revolution. Conditionally negative: Hitlerite Germany, Hitler, Jewish, Jewish.

The results of the study allow us to compare how the contextual environment of the same concept expressed by different concepts. It is not the same environment of his own name, associated with VI Lenin. Several nodal (parent) concepts were derived, which generated several contextual words. Thus, VI Lenin was represented in the textual array by the following parental concepts: "leader" (19 contextual links), "Vladimir Ilich" (107), "Leninism" (41), "Lenin monument" (73), "Lenin's images" (20), "Portrait of Lenin" (23), "Lenin's Mausoleum" (16), "Comrade Lenin" (28), "Ulyanov (Lenin)" (15), "Russophobe and the God-Bearer Lenin" (3).

For example, the environment of the concept "Vladimir Ilyich" is contradictory, and includes both positive speech markers ("happiness", "Russian", "speaker", "wonderful", etc.), and negative ("painful", "malicious", "Defeat", etc.).

The lively discussion of the monuments, portraits and images of Vladimir Lenin are related to contemporary events taking place in Ukraine and the destruction of monuments of the Soviet era, as well as a ban on communist symbols and general decommunization.

A separate structure of the concepts was devoted to the discussion of the attempt on Lenin. Obviously, this was due to the temporary structure of the text sample, which coincided with the dates of the assassination attempt - August 30, 1918 - which provoked discussion of this event 97 years later in social networks of Russia. So, several parents' concepts were dedicated to Fanny Kaplan.

The surge of discussions of the figure of Nicholas II in August and September is also related to the events of the shooting of the royal family in July 1918.

5 Conclusion

The paper represents the structural text analysis approach, which allows process large amount of unstructured text information, extract its thematic structure, extract and ranging its key concepts with theirs relationships. It can be useful in applications for example for social networks texts analysis, and its analysis in time dynamics. The instrument allows understand the latent structures of semantically co-related key concepts of text. The paper represents also the results of corpus texts analysis on 1917 year thematic. The results show that the social network content depends on the current social-political situation. The instrument allows explicit the context of actual concepts evaluation and validate the text corpus fragment choiced for analysis.

The publication is prepared within the supported by Russian Humanitarian Scientific Foundation the scientific project No. 15-36-12000 "Russia in 1917 in perception of modern youth: media discourse".

References

1. Blei, D.M.: Probabilistic topic models. *Commun. ACM* **55**(4), 77–84 (2012)
2. Dumais, S.T., Furnas, G.W., Landauer, T.K., Deerwester, S.: Using latent semantic analysis to improve information retrieval. In: Conference on Human Factors in Computing, pp. 281–285. ACM, New York (1988)
3. Hofmann, T.: Probabilistic latent semantic analysis. In: UAI 1999, pp. 289–296. Morgan Kaufmann Publishers Inc., San Francisco (1999)
4. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
5. Kharlamov, A.: Assotsiativnaya pamyat' - sreda dlya formirovaniya prostranstva znanii. Palmarium Academic Publishing, Dusseldorf (2017). (in Russian)
6. Kharlamov, A.A., Yermolenko, T.V., Zhonin, A.A.: Modeling of process dynamics by sequence of homogenous semantic networks on the base of text corpus sequence analysis. In: Ronzhin, A., Potapova, R., Delic, V. (eds.) SPECOM 2014. LNCS (LNAI), vol. 8773, pp. 300–307. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11581-8_37

Scientific Matchmaker: Collaborator Recommender System

Ilya Makarov^(✉) , Oleg Bulanov , Olga Gerasimova ,
Natalia Meshcheryakova , Ilia Karpov , and Leonid E. Zhukov 

National Research University Higher School of Economics,
Kochnovskiy Proezd 3, 125319 Moscow, Russia
iamakarov@hse.ru, revan1986@mail.ru

Abstract. Modern co-authorship networks contain hidden patterns of researchers interaction and publishing activities. We aim to provide a system for selecting a collaborator for joint research or an expert on a given list of topics. We have improved a recommender system for finding possible collaborator with respect to research interests and predicting quality and quantity of the anticipated publications. Our system is based on a co-authorship network derived from the bibliographic database, as well as content information on research papers obtained from SJR Scimago, staff information and the other features from the open data of researchers profiles. We formulate the recommendation problem as a weighted link prediction within the co-authorship network and evaluate its prediction for strong and weak ties in collaborative communities.

Keywords: Recommender systems · Co-authorship network
Scientific collaboration

1 Introduction

In a modern scientific community it is important to know the trends that provide the significant impact on the research fields. However, it is not easy to read hundreds of papers to become familiar with the new topics and small improvements in many related fields of study. The most natural way to select relevant and most valuable articles is by ordering a list of articles obtained by keywords query from some bibliography database according to a citation index or other centrality metrics for measuring simultaneously influence of the author and the paper on the respected research area [1]. In fact, such a method does not take into account the author professional skills, his respective research community and ability to publish his research at the international level. One of the first methods for selecting analyzing research community were made by Newman in [2, 3], where the author ordered authors according to the collaboration and centrality metrics in the co-authorship network.

Clustering approach for a co-authorship network of researchers who studied a particular disease was presented in [4]. The authors of [5] gave a representation of finance network analysis using similar methods. In [6, 7], the authors studied dependencies between citation indexes (predicted citations in [8]) and centralities in a co-authorship network. Data mining approaches for extracting significant features from the co-authorship networks specified to different research areas were presented in [2, 9]. Overall evaluation of methods and applications of network analysis were described in [10].

In this paper, we study a co-authorship network based on co-authorship network while one or more among the coauthors belong to the National Research University Higher School of Economics (HSE). The core idea of the research is to apply network analysis [10] and topic modelling [11] of research papers for the problem of extracting research interests of HSE workers and their respective skills based on the qualitative and quantitative data of their publications. Such obtained system then could be applied for automatising of expert search [12], building recommender system for searching a collaborator or scientific adviser [13], and simple search engine who could possess knowledge and skills related to a given description.

In what follows, we describe in details the process of evaluating work-in-progress recommender system based on co-authorship network and information on the staff units profiles of each author from HSE.

2 Data Processing and Problem Formalisation

As a starting point we took the database of all the records from the NRU HSE publication portal [14]. Here, we describe the process of cleaning original database:

1. All the duplicate records were merged under assumption that any conflict could be resolved by choosing the data from verified record at the portal.
2. All the missing fields were omitted during computational part of filed with median over respective category of articles and authors.
3. All the conflicts of different author First/Last Name representation were solved under simple logistic regression model based on the number of common co-authors in a given dataset.

After cleaning stage, we build five layers of the co-authorship network with authors as network actors and edges connecting authors with k jointly published research articles, $k = 1, \dots, 5$. The number 5 was chosen as the maximal number under which the network does not degenerate to the large number of small connected components of average sizes 1–2.

Next, we import features related to both, actors and ties between them. For each tie in the network representing co-authored publication, we added all the attributes of the publication the university portal [14] and imported subject areas and categories from Science Journal Ranking [15, 16]. We measure publication quality as its respective quartile in SJR ranking for the publication year,

computed as maximal (or average) over different categories per journal. In the future work we aim to include quartiles from the Web of Science Core Collection and provide unified algorithm of measuring relevant quartile in accordance to NRU HSE Science Fund Policy. One of the key features of new system will be choosing relevant quartile with respect to subject category choice over several research areas provided for indexed journal. Information about network actors, such as administrative unit position, declared author research interests and additional interests derived from topic modelling BigARTM system [17] over author's research publications were also included as node features.

We use the co-authorship network for various research problems related to collaboration patterns: dynamics of the number of papers as a personal progress indicator; dynamics of the number of collaborators as a representation of qualification and networking of the researcher; dynamics of the network communities density as an attribute to measure of team-working; impact of the research area and administrative units on collaborative and numerical publication patterns; using text mining, find a relevant expert or a collaborator based on his/her research interests and measured quality of co-author collaboration in international publications.

The recommender system gives a list of suggested candidates ordered by the inner metric of successful collaboration. More detailed, for a selected author the system generates a ranked list of authors whose papers could be relevant to him, and authors themselves could be good candidates for collaboration.

3 Measuring Similarity of Interests and Authors

We consider the problem of finding authors with similar interests to a selected one. In terms of network analysis, we study the problem of recommending similar author as a link prediction and use similarity between authors as model features. We choose well-known similarity scores described in [18].

We use *Common neighbors*, *Jaccard's coefficient*, *Adamic/Adar*, *Graph distance* similarity scores as baseline. In order to define similarity of actors by their known features from HSE staff information and publication activity represented by centralities of co-authorship network, we define additional content-based and graph-based features.

Nodes of the graph correspond to authors and, hence, have binary attributes, relation with NRU HSE and whether has administrative position, staff position, full-time status, and qualitative attributes, first name/last name, department hierarchy, and centrality metrics from the co-authorship network as metrics of influence. Edge attributes includes title and data of publication, journal quartile, weight, subject area and category, whether it was indexed by Scopus.

We computed cosine similarity for a vector consisting of normalized values of the feature parameters and "interests" metric as a normalized number of common journal SJR subject areas and categories for the journals, in which the original research article were published.

4 Training Model

In order to create proper training set for the person-based recommender system, we first need to select the list of potential candidates with similar interests.

We start with previous co-authors and staff members from the same HSE administrative unit. We construct feature vectors for their respective staff units by computing descriptive statistics based on research activity criteria and simple summarization of their respective researchers profiles related to different time intervals and qualities of publications.

We add to the list of candidates those administrative units with the difference between feature vectors less than the median of the distances between all the pairs of staff unit feature vectors representations, defined by the following features: the number of authors and papers, the ratio of the number of papers (from the department) by the ratio of the authors (from the department), the number of articles indexed in Scopus (over 3 last years) (published in Q1 and Q2 journal quartiles) (divided by the number of authors) (from the department), the average number of co-authors, the number of connected components, size of the greatest connected component (GCC), average distance, graph diameter, radius and density, average local clustering coefficient, the number of lecturers/senior lecturers/assistant professors/professors with their respective numbers of papers published (over the last 3 years), average rating of senior lecturers/assistant professors/professors from 0 to 30. The rating was calculated by the formula $\min(15 \cdot N, 30)/\min(15 \cdot N, 30)/\min(10 \cdot N, 30)/\min(6 \cdot N, 30)$, where N is the number of publications over the last 3 years. Each of the parameters was chosen according to either publishing activity criteria specified for different staff categories, or correlation between subgraph publishing patterns and their descriptive statistics.

In what follows, we use topic modelling for initial candidate research papers and find all the similar authors in HSE co-authorship network based on cosine metric of their research interests with the taxonomy obtained from hierarchical clustering of interests in respective co-occurrence network.

In order to catch structure of the network we used five methods of community detection on the co-authorship network: label propagation, fastgreedy, louvain, walktrap, infomap [19], and added candidates from the obtained clusters, to which original person belonged (Table 1).

The number of communities appears to be quite stable. Finally, we removed all non-HSE authors due to lack of information on their activity and status outside of HSE collaboration.

5 Recommender System

We used logistic regression with lasso regularization on normalized feature vectors to predict new links [20]. The parameter for regularization was chosen via model fitting by maximising accuracy of the model [21].

Table 1. The number of clusters obtained by different algorithm

Weight	>0	>1	>2	>3	>4	>5
Label	2265	1313	937	735	582	453
Fastgreedy	1120	954	748	608	511	406
Louvain	1102	943	744	605	509	406
Walktrap	2233	1270	890	673	542	428
Infomap	1988	1215	876	684	554	432

For a given researcher, we form subgraph of candidates from the previous section as a training set with the edges induced by original co-authorship network. We construct logistic regression model for each of the groups, taking as positive examples links in the chosen subgraph, and the same number of negative examples as missing links in order to keep the balance in the model, similar to [22].

We train our model on the “strong” co-authorship networks with each edge appearing only if up to $k = 2$ to $k = 5$ papers were written together and obtained a series of subgraphs to be used in evaluation (see [13]). For all the pairs of “weak” and “strong” subgraphs we prepare test set as links from the difference of these graphs and the same number of missing links from the links difference with features taken from the stronger subgraph. We calculated average error rates for test and train sets over all pairs of thresholds values of k (see Table 2). The area under the rock curve (AUC) and F1-measure are high, therefore, normalized lasso logistic regression was sufficient for binary classification.

Table 2. Similarity metrics

	Precision	Recall	Accuracy	F1-measure	AUC
Train data	0,93	0,99	0,96	0,96	0,99
Test data	0,91	0,87	0,91	0,90	0,94

6 Conclusion

We improved a recommender system [13] providing ranked list of candidates for collaboration based on HSE co-authorship network and database of publications. The recommender system demonstrates promising results on predicting new collaborations between existing authors and the accuracy of the system was improved by adding topic modelling component for extracting research interests from the original papers. The recommendations could also be made for a new author, who should state research interests and/or load his research papers for topics extraction.

We are looking forward to the evaluation of our system for several tasks inside the NRU HSE (though, it could be applied to any other research community), such as:

- finding an expert based on text for evaluation;
- matchmaking for co-authored research papers with novice researchers;
- searching for scientific adviser based on co-authorship network and the probability of publication in co-authorship with a student;
- searching for collaborators on specific grant proposal.

An application of this system may help stating the University policy to support novice researchers and increase their publishing activity or even, estimate collaboration between the University staff units.

Acknowledgements. The work was supported by the Russian Science Foundation under grant 17-11-01294 and performed at National Research University Higher School of Economics, Russia.

References

1. Liang, Y., Li, Q., Qian, T.: Finding relevant papers based on citation relations. In: Wang, H., Li, S., Oyama, S., Hu, X., Qian, T. (eds.) WAIM 2011. LNCS, vol. 6897, pp. 403–414. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23535-1_35
2. Newman, M.E.: Coauthorship networks and patterns of scientific collaboration. Proc. NAS **101**(suppl 1), 5200–5205 (2004)
3. Newman, M.: Who is the best connected scientist? A study of scientific coauthorship networks. Complex Netw. **6**(50), 337–370 (2004)
4. Morel, C.M., Serruya, S.J., Penna, G.O., Guimarães, R.: Co-authorship network analysis: a powerful tool for strategic planning of research, development and capacity building programs on neglected diseases. PLoS Negl. Trop. Dis. **3**(8), e501 (2009)
5. Cetorelli, N., Peristiani, S.: Prestigious stock exchanges: a network analysis of international financial centers. J. Bank. Finance **37**(5), 1543–1551 (2013)
6. Li, E.Y., Liao, C.H., Yen, H.R.: Co-authorship networks and research impact: a social capital perspective. Res. Policy **42**(9), 1515–1530 (2013)
7. Yan, E., Ding, Y.: Applying centrality measures to impact analysis: a coauthorship network analysis. J. IST Assoc. **60**(10), 2107–2118 (2009)
8. Sarigöl, E., et al.: Predicting scientific success based on coauthorship networks. EPJ Data Sci. **3**(1), 9 (2014)
9. Velden, T., Lagoze, C.: Patterns of collaboration in co-authorship networks in chemistry-mesoscopic analysis and interpretation. In: ISSI 2009 (2009)
10. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications, vol. 8. Cambridge University Press, Cambridge (1994)
11. Wang, C., Blei, D.M.: Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 448–456. ACM (2011)
12. Mimno, D., McCallum, A.: Expertise modeling for matching papers with reviewers. In: Proceedings of the 13th ACM SIGKDD IC, pp. 500–509 (2007)
13. Makarov, I., Bulanov, O., Zhukov, L.: Co-author recommender system. In: Kalyagin, V., Nikolaev, A., Pardalos, P., Prokopyev, O. (eds.) Springer Proceedings in Mathematics and Statistic, vol. 197, pp. 1–6. Springer, Cham (2017)

14. Powered by HSE Portal: Publications of HSE (2017). <http://publications.hse.ru/en>. Accessed 9 May 2017
15. González-Pereira, B., Guerrero-Bote, V.P., Moya-Anegón, F.: A new approach to the metric of journals' scientific prestige: the SJR indicator. *J. Informetr.* **4**(3), 379–391 (2010)
16. Guerrero-Bote, V.P., Moya-Anegón, F.: A further step forward in measuring journals' scientific prestige: the SJR2 indicator. *J. Informetr.* **6**(4), 674–688 (2012)
17. BigARTM contributors: BigARTM v0.8.2, December 2016. <https://doi.org/10.5281/zenodo.288960>
18. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. *J. IST Assoc.* **58**(7), 1019–1031 (2007)
19. Lancichinetti, A., Fortunato, S.: Community detection algorithms: a comparative analysis. *Phys. Rev. E* **80**(5), 056117 (2009)
20. Meier, L., Van De Geer, S., Bühlmann, P.: The group LASSO for logistic regression. *J. Roy. Stat. Soc. Ser. B (Stat. Methodol.)* **70**(1), 53–71 (2008)
21. Wainwright, M.J., Ravikumar, P., Lafferty, J.D.: High-dimensional graphical model selection using l_1 -regularized logistic regression. *Adv. Neural Inf. Process. Syst.* **19**, 1465 (2007)
22. Beel, J., et al.: Research paper recommender system evaluation: a quantitative literature survey. In: Proceedings of the International Workshop on RepSys 2013, pp. 15–22. ACM, New York (2013)



Erratum to: Bagging Prediction for Censored Data: Application for Theatre Demand

Evgeniy M. Ozhegov and Alina Ozhegova

Erratum to:

Chapter “Bagging Prediction for Censored Data: Application for Theatre Demand” in: W. M. P. van der Aalst et al. (Eds.): *Analysis of Images, Social Networks and Texts*, LNCS 10716, https://doi.org/10.1007/978-3-319-73013-4_18

The original version of the paper starting on p. 197 was revised. An acknowledgement has been added. The original article was corrected respectively.

The updated online version of this chapter can be found at
https://doi.org/10.1007/978-3-319-73013-4_18

Author Index

- Alimova, Ilseyar 3
Bakarov, Amir 16
Biemann, Chris 94
Bokovoy, Andrey 210
Bolshakova, Elena I. 22
Bryukhov, Anton 267
Bulanov, Oleg 404
Burnaev, Evgeny 245

Chershnev, Roman 131
Cherkasova, Galina 106
Chernoskutov, Mikhail 94

Dokuka, Sofia 381, 392
Durandin, Oleg 34

Eremeev, Anton V. 142
Eremeev, Sergey 172

Gerasimova, Olga 404
Gimadi, Edward Kh. 295, 304
Glavnov, Nikolay 164
Gradoselskaya, Galina 392
Gromova, Anna 152
Gureenkova, Olga 16

Ignatov, Dmitry I. 183
Ivanov, Kirill 59

Jung, Alexander 82

Kalenkova, Anna A. 371
Kapushev, Yermek 245
Karpov, Ilia 404
Kel'manov, Alexander V. 142, 313, 323
Kertész-Farkas, Attila 131
Khachay, Daniel 334

Khachay, Michael 334, 346
Khamidullin, Sergey 313
Khandeev, Vladimir 313
Kharchevnikova, Angelina S. 223
Kharlamov, Alexander 392
Kober, Vitaly 280
Kobylkin, Konstantin 356
Konchagin, Andrey M. 371
Krasnov, Fedor 164
Krekhovets, Ekaterina 381
Kunilovskaya, Maria 47
Kuptsov, Kirill 172
Kutuzov, Andrey 47
Kuznetsov, Andrey 231

Liao, Yiping 82
Litvyakov, Boris 183
Loukachevitch, Natalia 59

Ma, Lizhuang 255
Makarov, Ilya 183, 404
Malafeev, Alexey 34, 72
Malmi, Eric 82
Meshcheryakova, Natalia 404
Motkova, Anna 323
Myasnikov, Evgeny 237
Myasnikov, Vladislav 231

Neznakhina, Katherine 346
Nikolaev, Kirill 72
Nokel, Michael 59
Notchenko, Alexandr 245
Novskaya, Yulia 255

Ozhegov, Evgeniy M. 197
Ozhegova, Alina 197

Panchenko, Alexander 94
Philippovich, Yuriy 106
Poheriy, Maria 334
Pozdeev, Evgeniy 267

- Priymak, Margarita 381
Pyatkin, Artem V. 142
- Rodriges Zalipynis, Ramon Antonio 267
Romanov, Semyon 172
Ruchay, Alexey 280
Ruoqi, Sun 255
- Sapin, Alexander S. 22
Savchenko, Andrey V. 223
Savostyanov, Dmitry 183
Sayfullina, Luiza 82
Shcherbakov, Andrei 106
Shenmaier, Vladimir 323
Sitnikov, Alexander 164
Sokolova, Anastasiia D. 223
- Tsidulko, Oxana Yu. 304
Tutubalina, Elena 3
- Ustalov, Dmitry 94
- Vylomova, Ekaterina 106
- Yakovlev, Konstantin 210
- Zhu, Hengliang 255
Zhukov, Leonid E. 404
Ziegler, Igor A. 142
Zobnin, Alexey 116
Zolotykh, Nikolai 34