Peter Haber
Thomas Lampoltshammer
Manfred Mayr *Eds.*

Data Science – Analytics and Applications

Proceedings of the 1st International Data Science Conference – iDSC2017

EBOOK INSIDE

Springer Vieweg

Data Science – Analytics and Applications

Peter Haber · Thomas Lampoltshammer · Manfred Mayr
(Eds.)

Data Science – Analytics and Applications

Proceedings of the 1st International Data Science Conference – iDSC2017

Springer Vieweg

Editors
Peter Haber
Informationstechnik & System-Management
Fachhochschule Salzburg Puch/Salzburg, Österreich

Manfred Mayr
Informationstechnik & System-Management
Fachhochschule Salzburg Puch/Salzburg, Österreich

Thomas Lampoltshammer
Department für E-Governance in Wirtschaft und Verwaltung
Donau-Universität Krems, Krems an der Donau / Österreich

ISBN 978-3-658-19286-0 **ISBN 978-3-658-19287-7 (eBook)**
https://doi.org/10.1007/978-3-658-19287-7

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Springer Vieweg

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier

Springer Vieweg ist Teil von Springer Nature
Die eingetragene Gesellschaft ist Springer Fachmedien Wiesbaden GmbH
Die Anschrift der Gesellschaft ist: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Preface

It is with deep satisfaction that we write this foreword for the Proceedings of the 1st International Data Science Conference (iDSC) held in Salzburg, Austria, June 12th - 13th 2017. The conference program and the resulting proceedings represent the efforts of many people. We want to express our gratitude towards the members of our program committee as well as towards our external reviewers for their hard work during the reviewing process.

iDSC proofed itself as an innovative conference, which gave its participants the opportunity to delve into state-of-the-art research and best practice in the fields of Data Science and data-driven business concepts. Our research track offered a series of presentations by Data Science researchers regarding their current work in the fields of Data Mining, Machine Learning, Data Management, and the entire spectrum of Data Science.

In our industry track, practitioners demonstrated showcases of data-driven business concepts and how they use Data Science to achieve organisational goals, with a focus on manufacturing, retail, and financial services. Within each of these areas, experts described their experience, demonstrated their practical solutions, and provided an outlook into the future of Data Science in the business domain.

Besides these two parallel tracks, a European symposium on Text and Data Mining has been integrated into the conference. This symposium highlighted the EU project FutureTDM, granting insights into the future of Text and Data Mining, and introducing overarching policy recommendations and sector-specific guidelines to help stakeholders overcome the legal and technical barriers, as well the lack of skills that have been identified.

Our sponsors had their own, special platform via workshops to provide hands-on interaction with tools or to learn approaches towards concrete solutions. In addition, an exhibition of products and services offered by our sponsors took place throughout the conference, with the opportunity for our participants to seek contact and advice.

Completing the picture of our program, we proudly presented keynote presentations from leaders in Data Science and data-driven business, both researchers and practitioners. These keynotes provided all participants the opportunity to come together and shared views on challenges and trends in Data Science.

In addition to the contributed papers, five invited keynote presentations were given by: Euro Beinat (CS Research, Salzburg University), Mario Meir-Huber (Microsoft Austria), Mike Olson (Cloudera), Ralf Klinkenberg (RapidMiner) and Janek Strycharz (Digital Center Poland). We thank the invited speakers for sharing their insights with our community.

The conference chair John Thompson has also helped us in many ways setting up the industry track, for which we are grateful. We would especially like to thank our two colleagues, Astrid Karnutsch and Maximilian Tschuchnig, for their enormous and constructive commitment to organizing and conducting the conference. The paper submission and reviewing process was managed using the EasyChair system.

These proceedings will provide scientists and practitioners with an excellent reference to current activities in the Data Science domain. We trust also that this will be an impetus to stimulate further studies, research activities and applications in all discussed areas ensured by the support of our publisher Springer / Vieweg Wiesbaden Germany.

Finally, again, the conference would not be possible without the excellent papers contributed by our authors. We thank them for their contributions and their participation at iDSC´17.

Peter Haber, Thomas Lampoltshammer and Manfred Mayr

Conference Chairs

Future TDM Symposium Recap

FutureTDM is a european project focusing on reducing barriers and increasing uptake of Text and Data Mining (TDM) for research environments in Europe. The outcomes of the project were presented in the Symposium which has also served to connect key actors and interest groups and promote open dialogue via discussion panels and informal workshops. The FTDM Symposium was scheduled alongside iDSC 2017, given that both events address similar target groups and share a common perspective: they both aimed at creating a communication network among the members of the TDM community, where experts can exchange ideas and share the most up-to-date research results, as well as legal and industrial advances relevant to TDM. The audience targeted by the iDSC conference was the broad community of researchers and industry practitioners as well as other practitioners and stakeholders, making it ideal for disseminating the project's results.

The project's objective has been to detect the barriers to TDM, reveal best practices and put together sets of recommendations for TDM practitioners through a collaborative knowledge and open information approach. The barriers recorded were grouped around four pillars: a) **legal**, b) **economic,** c) **skills**, d) **technical**. These categories emerged after discussions with respective stakeholders such as researchers, developers, publishers and SMEs during Knowledge Cafés run across Europe (the Netherlands, the United Kingdom, Italy, Slovenia, Germany, Poland etc) and two workshops held in Brussels[1] (on September, 27th 2016 and March, 29th 2017).

The Symposium[2] was a chance to invite experts from all over Europe to share their experience and expertise in different domains. It was also a great opportunity to announce the guidelines and recommendations formulated in order to increase TDM uptake. It started with a brief introduction by Bernhard Jäger (SYNYO)[3] underlying the need to bring together different groups of stakeholders, such as policy makers and legislators, developers and users who would benefit from the project's findings and the respective recommendations formed by the FTDM working groups. It continued with a keynote speech by Janek Strycharz (Projekt Polska Foundation) dedicated to the Economic Potential of Data Analytics. Janek Strycharz elaborated on different types of Big Data and the variety of possibilities they offer and explained how that at a global and european scale there could be a benefit from Big Data and TDM (the European GDP alone would be increased by USD 200 billion).

[1] FutureTDM Workshop I and II outcomes can be found at http://www.futuretdm.eu/knowledge-cafes/futuretdm-workshop/
http://www.futuretdm.eu/knowledge-cafes/futuretdm-workshop-2/

[2] All presentation slides are available online at www.slideshare.net/FutureTDM/presentations

[3] Presentation on Introduction to the FutureTDM project is available at https://www.slideshare.net/FutureTDM/introduction-to-the-future-tdm-project

The first session entitled "**Data Analytics and the Legal Landscape: Intellectual Property and Data Protection"** included Freyja van den Boom, researcher from Open Knowledge International/Content Mine who presented the legal barriers identified and the respective recommendations created under the subject "Dealing with the legal bumps on the road to further TDM uptake". The focus of the presentation was on the principles identified to counterbalance barriers: Awareness and Clarity, TDM Without Boundaries, and Equitable Access. The session was chaired by Ben White (Head of Intellectual Property at the British Library) and included the following panelists: i) Duncan Campbell (John Wiley & Sons, Inc.), representing the publisher's perspective, ii) Prodromos Tsiavos (Onassis Cultural Centre/IP Advisor), providing an organization's point of view, iii) Marie Timmermann (Science Europe), offering her point of view as the EU Legislation and Regulatory Affairs Officer and iv) Romy Sigl (AustrianStartups) sharing her experience from startUps. The discussion revolved around regulations which must address the implementation of the law and its exceptions, copyright issues, the distinction between commercial and noncommercial activities, the need for better communication between different groups of stakeholders and the importance and value of TDM for publishers.

During the following session the projects ContentMine (Stefan Kasberger), PLAZI (Donat Agosti), CORE (Petr Knoth), RapidMiner (Ralf Klinkenberg), clarin:el (Maria Gavrilidou) and ALCIDE (Alessio Palmero Aprosio) were introduced and the presenters were accessible for a more detailed presentation of their work to the attendees who would be interested in learning more. The researchers shared their experience on technical and legal problems they have encountered demonstrating the TDM applications and infrastructures they had created.

The next session offered an **overview of FTDM case studies from Startups to Multinationals**. The presentation entitled "Stakeholder consultations - The Highlights" was given by Freyja van den Boom (Open Knowledge International/Content Mine) who talked about the findings from continuous stakeholder consultations throughout the project. The session was chaired by Maria Eskevich (Radboud University) and included as panelists Donat Agosti (PLAZI), Petr Knoth (CORE), Kim Nilsson (PIVIGO), and Peter Murray-Rust (ContentMine). The issues raised during discussion pinpointed the need for realistic solutions to infrastructures, community engagement, and open source and data.

Kiera McNeice (British Library) was the presenter in the fourth session and her presentation was entitled "Supporting TDM in the Education Sector". The session focusing on "**Universities, TDM and the need for strategic thinking on educating researchers**" was chaired by Ben White (Head of Intellectual Property at the British Library) and panelists Claire Sewell (Cambridge University Library), Jonas Holm (Stockholm University Library), and Kim Nilsson (PIVIGO). The discussion which followed touched upon issues such as the future of Data Science and the nature of Data Scientists. Some of the key concepts which were discussed were that of inclusion and diversity, gender imbalance and nationality characteristics, which all affect access to Data Science and the ability to become a Data Scientist. Concerns were expressed as to whether anyone could become a Data Scientist, and whether the focus should be on becoming a Data Scientist or a more efficient TDM user.

The challenges and solutions regarding technologies and infrastructures supporting Text and Data Analytics was the topic of the fifth session, the main presenter of which was Maria Eskevich (Radboud University). She focused on "The TDM Landscape: Infrastructure and Technical Implementation" and touched upon the business and scientific perspectives on TDM by showing the investment made by the EU in the five economic sectors. She also talked about the barriers/challenges encountered in terms of accessibility and interoperability of infrastructures, sustainability of data and digital readiness of language resources. The following discussion, chaired by Stelios Piperidis (ARC) with Mihai Lupu (Data Market Austria), Maria Gavrilidou (clarin: el) and Nelson Silva (know-centre) revolved around real TDM problems and the solutions the researchers came up with and close with the requirements of an effective TDM infrastructure.

The final session of the Symposium was dedicated to the **Next Steps: A Roadmap to promoting greater uptake of Data Analytics in Europe**. A presentation was made by Kiera McNeice (British Library) who briefly summarised what the project has achieved so far and focussed on the key principles from the FutureTDM Policy Framework[4] which must underlie all the efforts to be made in the future in Legal Policies, Skills and Education, Economy and Incentives and Technical and Infrastructure.

The Symposium close with a presentation of Bernhard Jäger and Burcu Akinci (SYNYO) of the FutureTDM platform (http://www.futuretdm.eu/), which is populated with the project outcomes and findings.The platform will continue to exist after the end of the project and will be continuously revised and updated in order to maintain a coherent and up-to-date view on the TDM landscape open to the public.

Kornella Pouli

Athena RIC/ILSP, Athens

Burcu Akinci

SYNYO GmbH, Vienna

[4] http://www.futuretdm.eu/policy-framework/

Organisation

Organising Institutions

Salzburg University of Applied Sciences
Information Professionals GmbH

Conference Chairs

Peter Haber	Salzburg University of Applied Sciences
Thomas J. Lampoltshammer	Danube University Krems
Manfred Mayr	Salzburg University of Applied Sciences
John A. Thompson	Information Professionals GmbH

Organising Committee

Peter Haber	Salzburg University of Applied Sciences
Astrid Karnutsch	Salzburg University of Applied Sciences
Thomas J. Lampoltshammer	Danube University Krems
Manfred Mayr	Salzburg University of Applied Sciences
John A. Thompson	Information Professionals GmbH
Susanne Schnitzer	Information Professionals GmbH
Maximilian E. Tschuchnig	Salzburg University of Applied Sciences

Program Committee

David C. Anastasiu	San Jose State University
Vera Andrejcenko	University of Antwerp
Christian Bauckhage	University of Bonn
Markus Breunig	Rosenheim University of Applied Sciences
Stefanie Cox	IT Innovation Centre
Werner Dubitzky	University of Ulster, Coleraine
Günther Eibl	Salzburg University of Applied Sciences
Süleyman Eken	University Kocaeli
Karl Entacher	Salzburg University of Applied Sciences
Edison Pignaton de Freitas	Federal University of Rio Grande do Sul
Bernhard Geissler	Danube University Krems
Charlotte Gerritsen	Netherlands Institute for the Study of Crime and Law Enforcement (NSCR)
Mohammad Ghoniem	Luxembourg Institute of Science and Technology
Peter Haber	Salzburg University of Applied Sciences
Johann Höchtl	Danube University Krems
Martin Kaltenböck	Semantic Web Company
Astrid Karnutsch	Salzburg University of Applied Sciences
Elmar Kiesling	Vienna University of Technology
Robert Krimmer	University of Tallinn
Peer Kröger	Ludwig-Maximilians-Universität München
Thomas J. Lampoltshammer	Danube University Krems

Michael Leitner	Louisiana State University
Giuseppe Manco	University of Calabria
Manfred Mayr	Salzburg University of Applied Sciences
Mark-David McLaughlin	Bentley University
Robert Merz	Salzburg University of Applied Sciences
Elena Lloret Pastor	University of Alicante
Cody Ryan Peeples	Cisco
Gabriela Viale Pereira	Fundação Getúlio Vargas – EAESP
Peter Ranacher	University of Zurich
Siegfried Reich	Salzburg Research Forschungsgesellschaft mbH
Eric Rozier	Iowa State University
Johannes Scholz	Graz University of Technology
Maximilian E. Tschuchnig	Salzburg University of Applied Sciences
Jürgen Umbrich	Vienna University of Economics and Business
Andreas Unterweger	Salzburg University of Applied Sciences
Eveline Wandl-Vogt	Austrian Academy of Sciences
Stefan Wegenkittl	Salzburg University of Applied Sciences
Stefanie Wiegand	IT Innovation Centre / University of Southampton
Peter Wild	Austrian Institute of Technology
Radboud Winkels	University of Amsterdam
Anneke Zuiderwijk - van Eijk	Delft University of Technology

Reviewer

David C. Anastasiu	San Jose State University
Christian Bauckhage	University of Bonn
Markus Breunig	Rosenheim University of Applied Sciences
Cornelia Ferner	Salzburg University of Applied Sciences
Werner Dubitzky	University of Ulster, Coleraine
Günther Eibl	Salzburg University of Applied Sciences
Karl Entacher	Salzburg University of Applied Sciences
Bernhard Geissler	Danube University Krems Höchtl
Martin Kaltenböck	Semantic Web Company
Peer Kröger	Ludwig-Maximilians-Universität München
Thomas J. Lampoltshammer	Danube University Krems
Michael Leitner	Louisiana State University
Elena Lloret Pastor	University of Alicante
Manfred Mayr	Salzburg University of Applied Sciences
Robert Merz	Salzburg University of Applied Sciences
Edison Pignaton de Freitas	Federal University of Rio Grande do Sul
Siegfried Reich	Salzburg Research Forschungsgesellschaft mbH
Eric Rozier	Iowa State University
Johannes Scholz	Graz University of Technology
Maximilian E. Tschuchnig	Salzburg University of Applied Sciences
Jürgen Umbrich	Vienna University of Economics and Business
Andreas Unterweger	Salzburg University of Applied Sciences
Stefan Wegenkittl	Salzburg University of Applied Sciences

Sponsors of the conference

Platinum Sponsors

Cloudera GmbH
Apache Hadoop-based software, support and services, and training

www.cloudera.com

Silver Sponsors

The unbelievable Machine Company GmbH
Full-service provider for Big Data, cloud services & hosting

www.unbelievable-machine.com

F&F GmbH
IT consulting, solutions and Big Data Analytics

www.ff-muenchen.de

The MathWorks GmbH
Mathematical computing software

www.mathworks.com

RapidMiner GmbH
Data science software platform for data preparation, machine learning, deep learning, text mining, and predictive analytics

www.rapidminer.com

ITG: innovative consulting and location development
ITG is Salzburg's innovation centre

www.itg-salzburg.at

Table of Content

German Abstracts

Reasoning and Predictive Analytics

Circadian Cycles and Work Under Pressure: A Stochastic Process Model for E-learning Population Dynamics

Internetanalysetechniken, konzipiert zur Quantifizierung von Internetnutzungsmustern, erlauben ein tieferes Verständnis menschlichen Verhaltens. Neueste Modelle menschlicher Verhaltensdynamiken haben gezeigt, dass im Gegensatz zu zufällig verteilten Ereignissen, Menschen Tätigkeiten ausüben, die schubweises Verhalten aufweisen. Besonders die Teilnahme an Internetkursen zeigt häufig Zeiträume von Inaktivität und Prokrastination gefolgt von häufigen Besuchen kurz vor den Prüfungen. Hier empfehlen wir ein stochastisches Prozessmodell, welches solche Muster kennzeichnet und Tagesrhythmen menschlicher Aktivitäten einbezieht. Wir bewerten unser Modell anhand von realen Daten, die während einer Zeitspanne von zwei Jahren auf einer Plattform für Universitätskurse gesammelt wurden. Anschließend schlagen wir ein dynamisches Modell vor, welches sowohl Prokrastinationszeiträume als auch Zeiträume des Arbeitens unter Zeitdruck berücksichtigt. Da Tagesrhythmen und Prokrastination-Druck-Kreisläufe wesentlich für menschliches Verhalten sind, kann unsere Methode auf andere Tätigkeiten ausgeweitet werden, wie zum Beispiel die Auswertung von Surfgewohnheiten und Kaufverhalten von Kunden.

Investigating and Forecasting User Activities in Newsblogs: A Study of Seasonality, Volatility and Attention Burst

Das Studium allgemeiner Aufmerksamkeit ist ein Hauptthemengebiet im Bereich der Internetwissenschaft, da wir wissen wollen, wie die Beliebtheit eines bestimmten Nachrichtenthemas oder Memes im Laufe der Zeit zu- oder abnimmt. Neueste Forschungen konzentrierten sich auf die Entwicklung von Methoden zur Quantifizierung von Erfolg und Beliebtheit von Themen und untersuchten ihre Dynamiken im Laufe der Zeit. Allerdings wurde das gesamtheitliche Nutzerverhalten über Inhaltserstellungsplattformen größtenteils ignoriert, obwohl die Beliebtheit von Nachrichtenartikeln auch mit der Art verbunden ist, wie Nutzer Internetplattformen nutzen. In dieser Abhandlung zeigen wir ein neuartiges Framework, dass die Verlagerung der Aufmerksamkeit von Bevölkerungsgruppen in Hinblick auf Nachrichtenblogs untersucht. Wir konzentrieren uns auf das Kommentarverhalten von Nutzern bei Nachrichtenbeiträgen, was als Stellvertreter für die Aufmerksamkeit gegenüber Internetinhalten fungiert. Wir nutzen Methoden der Signalverarbeitung und Ökonometrie, um Verhaltensmuster von Nutzern aufzudecken, die es uns dann erlauben, das Verhalten einer Bevölkerungsgruppe zu simulieren und schlussendlich vorherzusagen, sobald eine Aufmerksamkeitsverlagerung auftritt. Nach der Untersuchung von Datenreihen von über 200 Blogs mit 14 Millionen Nachrichtenbeiträgen, haben wir zyklische Gesetzmäßigkeiten im Kommentarverhalten identifiziert: Aktivitätszyklen von 7 Tagen und 24 Tagen, die möglicherweise im Zusammenhang zu bekannten Dimensionen von Meme-Lebenszeiten stehen.

Knowledge-based Short-Term Load Forecasting for Maritime Container Terminals

Durch den Anstieg von Last- und Nachfragemanagement in modernen Energiesystemen erhält die Kurzzeitlastprognose für Industrieeinzelkunden immer mehr Aufmerksamkeit. Es scheint für Industriestandorte lohnenswert, dass Wissen zu geplanten Maßnahmen in den Lastprognoseprozess des nächsten Tages einzubeziehen. Im Fall eines Seecontainer-Terminals, basieren diese Betriebspläne auf der Liste ankommender und abfahrender Schiffe. In dieser Abhandlung werden zwei Ansätze vorgestellt, welche dieses Wissen auf verschiedene Weisen einbeziehen: Während fallbasiertes Schlussfolgern träge während des Prognoseprozesses lernt, müssen künstliche neurale Netzwerke erst trainiert werden, bevor ein Prognoseprozess durchgeführt werden kann. Es kann gezeigt werden, dass das Einbeziehung von mehr Wissen in den Prognoseprozess bessere Ergebnisse im Hinblick auf die Prognosegenauigkeit ermöglicht.

Data Analytics in Community Networks

Beyond Spectral Clustering: A Comparative Study of Community Detection for Document Clustering

Dokumenten-Clustering ist ein allgegenwärtiges Problem bei der Datengewinnung, da Textdaten eine der gebräuchlichsten Kommunikationsformen sind. Die Reichhaltigkeit der Daten erfordert Methoden, die – je nach den Eigenschaften der Informationen, die gewonnen werden sollen – auf verschiedene Aufgaben zugeschnitten sind. In letzter Zeit wurden graphenbasierte Methoden entwickelt, die es hierarchischen, unscharfen und nicht-gaußförmigen Dichtemerkmalen erlauben, Strukturen in komplizierten Datenreihen zu identifizieren. In dieser Abhandlung zeigen wir eine neue Methodologie für das Dokumenten-Clustering, das auf einem Graphen basiert, der durch ein Vektorraummodell definiert ist. Wir nutzen einen überlappenden hierarchischen Algorithmus und zeigen die Gleichwertigkeit unserer Qualitätsfunktion mit der von Ncut. Wir vergleichen unsere Methode mit spektralem Clustering und anderen graphenbasierten Modellen und stellen fest, dass unsere Methode eine gute und flexible Alternative für das Nachrichten-Clustering darstellt, wenn eingehende Details zwischen den Themen benötigt werden.

Third Party Effect: Community Based Spreading in Complex Networks

Ein wesentlicher Teil der Netzwerkforschung wurde dem Studium von Streuprozessen und Gemeinschaftserkennung gewidmet, ohne dabei die Rolle der Gemeinschaften bei den Merkmalen der Streuprozesse zu berücksichtigen. Hier verallgemeinern wir das SIR-Modell von Epidemien durch die Einführung einer Matrix von Gemeinschaftsansteckungsraten, um die heterogene Natur des Streuens zu erfassen, die durch die natürlichen Merkmale von Gemeinschaften definiert sind. Wir stellen fest, dass die Streufähigkeiten einer Gemeinschaft gegenüber einer anderen durch das interne Verhalten von Drittgemeinschaften beeinflusst wird. Unsere Ergebnisse bieten Einblicke in Systeme mit reichhaltigen Informationsstrukturen und in Populationen mit vielfältigen Immunreaktionen.

Cosine Approximate Nearest Neighbors

Kosinus-Ähnlichkeitsgraphenerstellung, oder All-Pairs-Ähnlichkeitssuche, ist ein wichtiger Systemkern vieler Methoden der Datengewinnung und des maschinellen Lernens. Die Graphenerstellung ist eine schwierige Aufgabe. Bis zu n2 Objektpaare sollten intuitiv verglichen werden, um das Problem für eine Reihe von n Objekten zu lösen. Für große Objektreihen wurden Näherungslösungen für dieses Problem vorgeschlagen, welche die Komplexität der Aufgabe thematisieren, indem die meisten, aber nicht unbedingt alle, nächsten Nachbarn abgefragt werden. Wir schlagen eine neue Näherungsgraphen-Erstellungsmethode vor, welche Eigenschaften der Objektvektoren kombiniert, um effektiv weniger Vergleichskandidaten auszuwählen, welche wahrscheinlich Nachbarn sind. Außerdem kombiniert unsere Methode Filterstrategien, welche vor kurzem entwickelt wurden, um Vergleichskandidaten, die nicht vielversprechend sind, schnell auszuschließen, was zu weniger allgemeinen Ähnlichkeitsberechnungen und erhöhter Effizienz führt. Wir vergleichen unsere Methode mit mehreren gängigen Annäherungs- und exakten Grundwerten von sechs Datensätzen aus der Praxis. Unsere Ergebnisse zeigen, dass unser Ansatz einen guten Kompromiss zwischen Effizienz und Effektivität darstellt, mit einer 35,81-fachen Effizienzsteigerung gegenüber der besten Alternative bei 0,9 Recall.

Data Analytics through Sentiment Analysis

Information Extraction Engine for Sentiment-Topic Matching in Product Intelligence Applications

Produktbewertungen sind eine wertvolle Informationsquelle sowohl für Unternehmen als auch für Kunden. Während Unternehmen diese Informationen dazu nutzen, ihre Produkte zu verbessern, benötigen Kunden sie als Unterstützung für die Entscheidungsfindung. Mit Bewertungen, Kommentaren und zusätzlichen Informationen versuchen viele Onlineshops potenzielle Kunden dazu zu animieren, auf ihrer Seite einzukaufen. Allerdings mangelt es aktuellen Online-Bewertungen an einer Kurzzusammenfassung, inwieweit bestimmte Produktbestandteile den Kundenwünschen entsprechen, wodurch der Produktvergleich erschwert wird. Daher haben wir ein Produktinformationswerkzeug entwickelt, dass gängige Technologien in einer Engine maschineller Sprachverarbeitung vereint. Die Engine ist in der Lage produktbezogene Online-Daten zu sammeln und zu sichern, Metadaten auszulesen und Meinungen. Die Engine wird auf technische Online-Produktbewertungen zur Stimmungsanalyse auf Bestandteilsebene angewendet. Der vollautomatisierte Prozess durchsucht das Internet nach Expertenbewertungen, die sich auf Produktbestandteile beziehen, und aggregiert die Stimmungswerte der Bewertungen.

Towards German Word Embeddings: A Use Case with Predictive Sentiment Analysis

Trotz des Forschungsbooms im Bereich Worteinbettungen und ihrer Textmininganwendungen der letzten Jahre, konzentriert sich der Großteil der Publikationen ausschließlich auf die englische Sprache. Außerdem ist die Hyperparameterabstimmung ein Prozess, der selten gut dokumentiert (speziell für nicht-englische Texte), jedoch sehr wichtig ist, um hochqualitative Wortwiedergaben zu erhalten. In dieser Arbeit zeigen wir, wie verschiedene Hyperparameterkombinationen Einfluss auf die resultierenden deutschen Wortvektoren haben und wie diese Wortwiedergaben Teil eines komplexeren Modells sein können. Im Einzelnen führen wir als erstes eine intrinsische Bewertung unserer deutschen Worteinbettungen durch, die später in einem vorausschauenden Stimmungsanalysemodell verwendet werden. Letzteres dient nicht nur einer intrinsischen Bewertung der deutschen Worteinbettungen, sondern zeigt außerdem, ob Kundenwünsche nur durch das Einbetten von Dokumenten vorhergesagt werden können.

User/Customer-centric Data Analytics

Feature Extraction and Large Activity-Set Recognition Using Mobile Phone Sensors

Diese Arbeit beschäftigt sich mit dem Problem der Aktivitätserkennung unter Verwendung von Daten, die vom Mobiltelefon des Benutzers erhoben wurden. Wir beginnen mit der Betrachtung und Bewertung der Beschränkungen der gängigen Aktivitätserkennungsansätze für Mobiltelefone. Danach stellen wir unseren Ansatz zur Erkennung einer großen Anzahl von Aktivitäten vor, welche die meisten Nutzeraktivitäten abdeckt. Außerdem werden verschiedene Umgebungen unterstützt, wie zum Beispiel zu Hause, auf Arbeit und unterwegs. Unser Ansatz empfiehlt ein einstufiges Klassifikationsmodell, dass die Aktivitäten genau klassifiziert, eine große Anzahl von Aktivitäten umfangreich abdeckt und in realen Umgebungen umsetzbar anzuwenden ist. In der Literatur gibt es keinen einzigen Ansatz, der alle drei Eigenschaften in sich vereint. In der Regel optimieren vorhandene Ansätze ihre Modelle entweder für einen oder maximal zwei der folgenden Eigenschaften: Genauigkeit, Umfang und Anwendbarkeit. Unsere Ergebnisse zeigen, dass unser Ansatz ausreichende Leistung im Hinblick auf Genauigkeit bei einem realistischen Datensatz erbringt, trotz deutlich erhöhter Aktivitätszahl im Vergleich zu gängigen Modellen, die auf Aktivitätserkennen basieren.

The Choice of Metric for Clustering of Electrical Power Distribution Consumers

Ein bedeutender Teil jedes Systemdatenmodells zur Energieverteilungsverwaltung ist ein Modell der Belastungsart. Eine Belastungsart stellt ein typisches Belastungsverhalten einer Gruppe gleicher Kunden dar, z. B. einer Gruppe von Haushalts-, Industrie- oder gewerblichen Kunden. Eine verbreitete Methode der Erstellung von Belastungsarten ist die Bündelung individueller Energieverbraucher auf der Basis ihres jährlichen Stromverbrauchs. Um ein zufriedenstellendes Maß an Belastungsartqualität zu erreichen, ist die Wahl des geeigneten Ähnlichkeitsmaßes zur Bündelung entscheidend. In dieser Abhandlung zeigen wir einen Vergleich verschiedener Metriken auf, die als Ähnlichkeitsmaß in unserem Prozess der Belastungsarterstellung eingesetzt werden. Zusätzlich zeigen wir eine neue Metrik, die auch im Vergleich enthalten ist. Die Metriken und die Qualität der damit erstellten Belastungsarten werden unter Verwendung von Realdatensätzen untersucht, die über intelligente Stromzähler des Verteilungsnetzes erhoben wurden.

Evolution of the Bitcoin Address Graph

Bitcoin ist eine dezentrale virtuelle Währung, die dafür genutzt werden kann, weltweit pseudo-anonymisierte Zahlungen innerhalt kurzer Zeit und mit vergleichsweise geringen Transaktionskosten auszuführen. In dieser Abhandlung zeigen wir die ersten Ergebnisse eine Langzeitstudie zur Bitcoinadressenkurve, die alle Adressen und Transaktionen seit dem Start von Bitcoin im Januar 2009 bis zum 31. August 2016 enthält. Unsere Untersuchung enthüllt eine stark verschobene Gradverteilung mit einer geringen Anzahl von Ausnahmen und zeigt, dass sich die gesamte Kurve stark ausdehnt. Außerdem zeigt sie die Macht der Adressbündelungsheuristik zur Identifikation von realen Akteuren, die es bevorzugen, Bitcoin für den Wertetransfer statt für die Wertespeicherung zu verwenden. Wir gehen davon aus, dass diese Abhandlung neue Einblicke in virtuelle Währungsökosysteme bietet und als Grundlage für das Design zukünftiger Untersuchungsmethoden und -infrastrukturen dienen kann.

Data Analytics in Industrial Application Scenarios

A Reference Architecture for Quality Improvement in Steel Production

Es gibt weltweit einen erhöhten Bedarf an Stahl, aber die Stahlherstellung ist ein enorm anspruchsvoller und kostenintensiver Prozess, bei dem gute Qualität schwer zu erreichen ist. Die Verbesserung der Qualität ist noch immer die größte Herausforderung, der sich die Stahlbranche gegenüber sieht. Das EU-Projekt PRESED (Predictive Sensor Data Mining for Product Quality Improvement) [Vorrausschauende Sensordatengewinnung zur Verbesserung der Produktqualität] stellt sich dieser Herausforderung durch die Fokussierung auf weitverbreitete, wiederkehrende Probleme. Die Vielfalt und Richtigkeit der Daten sowie die Veränderung der Eigenschaften des untersuchten Materials erschwert die Interpretation der Daten. In dieser Abhandlung stellen wir die Referenzarchitektur von PRESED vor, die speziell angefertigt wurde, um die zentralen Anliegen der Verwaltung und Operationalisierung von Daten zu thematisieren. Die Architektur kombiniert große und intelligente Datenkonzepte mit Datengewinnungsalgorithmen. Datenvorverarbeitung und vorausschauende Analyseaufgaben werden durch ein plastisches Datenmodell unterstützt. Der Ansatz erlaubt es den Nutzern, Prozesse zu gestalten und mehrere Algorithmen zu bewerten, die sich gezielt mit dem vorliegenden Problem befassen. Das Konzept umfasst die Sicherung und Nutzung vollständiger Produktionsdaten, anstatt sich auf aggregierte Werte zu verlassen. Erste Ergebnisse der Datenmodellierung zeigen, dass die detailgenaue Vorverarbeitung von Zeitreihendaten durch Merkmalserkennung und Prognosen im Vergleich zu traditionell verwendeter Aggregationsstatistik überlegene Erkenntnisse bietet.

Anomaly Detection and Structural Analysis in Industrial Production Environments

Das Erkennen von anormalem Verhalten kann im Kontext industrieller Anwendung von entscheidender Bedeutung sein. Während moderne Produktionsanlagen mit hochentwickelten Alarmsteuerungssytemen ausgestattet sind, reagieren diese hauptsächlich auf Einzelereignisse. Aufgrund der großen Anzahl und der verschiedenen Arten von Datenquellen ist ein einheitlicher Ansatz zur Anomalieerkennung nicht immer möglich. Eine weitverbreitete Datenart sind Logeinträge von Alarmmeldungen. Sie erlauben im Vergleich zu Sensorrohdaten einen höheren Abstraktionsgrad. In einem industriellen Produktionsszenario verwenden wir sequentielle Alarmdaten zur Anomalieerkennung und -auswertung, basierend auf erstrangigen Markov-Kettenmodellen. Wir umreißen hypothesegetriebene und beschreibungsorientierte Modellierungsoptionen. Außerdem stellen wir ein interaktives Dashboard zur Verfügung, um die Ergebnisse zu untersuchen und darzustellen.

Semantically Annotated Manufacturing Data to support Decision Making in Industry 4.0: A Use-Case Driven Approach

Intelligente Fertigung oder Industrie 4.0 ist ein Schlüsselkonzept, um die Produktivität und Qualität in industriellen Fertigungsunternehmen durch Automatisierung und datengetriebene Methoden zu erhöhen. Intelligente Fertigung nutzt Theorien cyber-physischer Systeme, dem Internet der Dinge sowie des Cloud-Computing. In dieser Abhandlung konzentrieren sich die Autoren auf Ontologie und (räumliche) Semantik, die als Technologie dienen, um semantische Kompatibilität der Fertigungsdaten sicherzustellen. Zusätzlich empfiehlt die Abhandlung, fertigungsrelevante Daten über die Einführung von Geografie und Semantik als Sortierformate zu strukturieren. Der in dieser Abhandlung verfolgte Ansatz sichert Fertigungsdaten verschiedener IT-Systeme in einer Graphdatenbank. Während des Datenintegrationsprozesses kommentiert das System systematisch die Daten – basierend auf einer Ontologie, die für diesen Zweck entwickelt wurde – und hängt räumliche Informationen an. Der in dieser Abhandlung vorgestellte Ansatz nutzt eine Analyse von Fertigungsdaten in Bezug auf Semantik und räumliche Abmessung. Die Methodologie wird auf zwei Anwendungsfälle für ein Halbleiterfertigungsunternehmen angewendet. Der erste Anwendungsfall behandelt die Datenanalyse zur Ereignisanalyse unter Verwendung von semantischen Ähnlichkeiten. Der zweite Anwendungsfall unterstützt die Entscheidungsfindung in der Fertigungsumgebung durch die Identifizierung potentieller Engpässe bei der Halbleiterfertigungslinie.

Full Papers – Double Blind Reviewed

Reasoning and Predictive Analytics

Circadian Cycles and Work Under Pressure: A Stochastic Process Model for E-learning Population Dynamics

Christian Backhage, César Ojeda and Rafet Sifa
Fraunhofer IAIS
St. Augustin, Germany

Christian Backhage and Rafet Sifa
University of Bonn
Bonn, Germany

***Abstract*—Web analytics techniques designed to quantify Web usage patterns allow for a deeper understanding of human behavior. Recent models of human behavior dynamics have shown that, in contrast to randomly distributed events, people engage in activities which show bursty behavior. In particular, participation in online courses often shows periods of inactivity and procrastination followed by frequent visits shortly before examination deadlines. Here, we propose a stochastic process model which characterizes such patterns and incorporates circadian cycles of human activities. We validate our model against real data spanning two years of activity on a university course platform. We then propose a dynamical model which accounts for both periods of procrastination and work under pressure. Since circadian and procrastination-pressure cycles are fundamental to human activities, our method can be extended to other tasks such as analyzing browsing behaviors or customer purchasing patterns.**

I. Introduction

Over the past few years, several platforms for open online courses have been launched that cater to the demand for continuous learning in the knowledge society. Benefits of these systems are that they allow a single professor to reach thousands of students, facilitate personalized learning, and, last but not least, allow for gathering information as to the behavior of large populations of students. The latter enables to track the learning progress and to automatically recommend content so as to optimize the learning experience.

Among the many data collected, visitation patterns play an especially crucial role and were found to reflect priority based decision making behaviors of human agents [1]–[4]. Such decision making phenomena are especially characteristic for online courses on university eLearning platforms where students are provided with the course material and are expected to cover it over the period of a semester. At the same time, research on online communication patterns has shown that inter-event times can be characterized in terms of inhomogeneous Poisson processes where the Poisson rate λ changes over time so as to account for the circadian cycles and weekly cycles in human activity [5]. In this paper, we therefore consider the use of such models in analyzing the behavior dynamics of a population engaged in online courses. We extend the inhomogeneous Poisson process and incorporate a dynamic equation that accounts for the sudden change in attention as the population reacts to a given deadline.

A. Empirical Basis

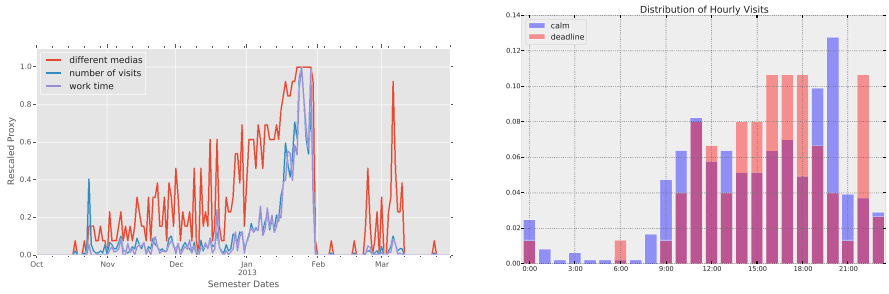
The empirical basis for our work in this paper consists of population behavior data collected from a course management system of an anonymous German university. In total, the system provides access to 1,147 different online lectures from 115 different courses. Our data set captures a total of 186,658 anonymized, time-stamped visits from 30,497 different IP addresses covering the four semesters in the time from April 2012 to March 2014. For each course covered in our data, students attend weekly lectures and exercises. Whenever a deadline for course work is scheduled, we typically observe students to react in terms of increasingly frequent visits to the course site. Immediately after each deadline, however, access counts drop significantly and this pattern tends to persist throughout the duration of the course. In addition to behaviors induced by course specific deadlines, we observe a fluctuating visiting rate caused by the personal schedules of students. Finally, prior to the final examinations, we typically observe highly frequent visits to course sites where, for some courses, the time in which students react to the final deadline spans several weeks while, for other courses, it is of the order of days. From an abstract point of view, increased visits prior to deadlines for course work and examination provide an example of a well known decision based queuing process [1] since deadlines cause priorities of students to shift. Looking at our data on a finer, say, daily level, we can observe how students allocate time to common activities such as eating, resting, or sleeping. These natural activities cause idle periods in our data where site visitation rates drop and we note that such periods cannot be explained in terms of simple Poisson processes.

The examples in Fig. 1 illustrate these general behaviors. In particular, the figure shows three proxies: the number of visits to a course site, time spent on the course (working time in seconds), and number of different media (videos, course notes, ...) viewed per day. For better comparison, we rescaled each proxy to the same maximum value. Apparently, these proxies only vary minimally in the diurnal cycles from lecture to deadline periods.

B. Contributions

Addressing the problem of modeling the population behavior on eLearning sites, our main contributions in this paper is to introduce a model of human behavior dynamics. In

P. Haber et al. (Hrsg.), *Data Science – Analytics and Applications*, http://doi.org/10.1007/978-3-658-19287-7_1



(a) distribution of proxies over a semester

(b) average distribution of visits per day

Fig. 1. Example of the temporal distribution of several proxies for the activity on a course related site in an eLearning system.

the following, we will refer to this model as a *Pairwise Procrastination Reaction Cascade* (PPRC) which that allows for answering the following questions: a) Given a particular university course, can we describe the inter-event time distribution of visits to the system? b) Given a course, can we predict from the behavior observed in the initial stages of the reaction period, whether a population will covert most of the material required?

II. Model Definition

Our goal is to model the visiting behavior of a population of students to a video lectures platform over the period of a semester in which students access course content which consists of video lectures and reading material. In particular, we aim at representing the behavior in the time span between the beginning of the semester and the final examination. Should a course require several examinations, we model the behavior over the period prior to the "most important examination" (defined by the amount of course content to be covered). We assume that there are N_u different users and that their number remains fixed during a semester. In order to be of practical use, our model should account for:

1) circadian cycles of human activity
2) the stochastic nature of aggregated behaviors of many different users
3) the tendency of students to learn most of the material close to the deadline
4) the short term behavior of a session of study

A. Inhomogeneous Poisson Process

The visiting pattern of the population is defined by a set of point data in a one dimensional domain $\mathcal{S} \in \mathbb{R}$. The elements of $\mathcal{S}$ will be called t_i and indicate the number of seconds which elapsed from the beginning of the semester to the point in time at which visit $i \in \{1, ..., N\}$ occurs where N is the total number of visits in the semester for the course we are trying to model.

Poisson processes are widely used for modeling point data. In particular, if the rate of arrivals change over time, inhomogeneous Poisson process allow for a variation in the arrivals rate, defined trough an intensity function

$$\lambda(t) : \mathcal{S} \to \mathbb{R}^{+}. \tag{1}$$

This intensity function contains information about the visiting behavior, because the probability of the number of visits between time t and time $t + \delta$, $Pr\{\tilde{N}(t, t + \delta)\}$ must, by definition [6], satisfy

$$Pr\Big\{\tilde{N}(t, t + \delta) = 0\Big\} = 1 - \delta\lambda(t) - o(\delta) \tag{2}$$

$$Pr\Big\{\tilde{N}(t, t + \delta) = 1\Big\} = \delta\lambda(t) + o(\delta) \tag{3}$$

for all $t \geq 0$ and some vanishingly small δ . The probability of a visit to occur at time t then depends on the value of λ at t and, by imposing conditions on λ, one can devise inhomogeneous Poisson process models of a wide variety of behaviors. In the following, we incorporate the known empirical behavior by characterizing the intensity function.

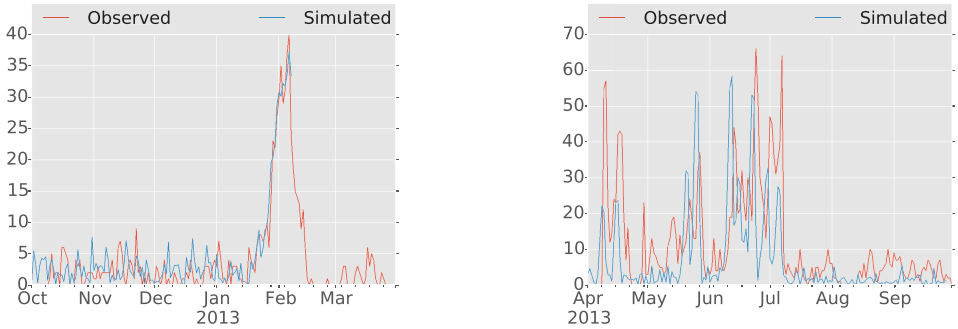
B. Circadian Cycles

As seen in Fig. 1(b), circadian cycles influence human behavior over the course of a day and, not surprisingly, lead to reduced visits to eLearning sites during night times.

We incorporate this kind of prior knowledge using empirically determined histograms of hourly visits as shown in the figure and define a function $P_d(t)$ which indicates the probability of a visit to occur at time t (we need to evaluate the hour of the day for the value of t). We define P_d to be periodic, $P_d(t + \tau_d) = P_d(t)$ where the period τ_d corresponds to the number of seconds in a day. Given these preliminaries, we can then incorporate P_d in the behavior of the Poisson rate using

$$\lambda(t) = V_d(t)P_d(t) \tag{4}$$

where $V_d(t)$ indicates the rate of visits per day. Next, we introduce a dynamical model for this rate.



(a) Procrastination and Reaction

(b) Piecewise Procrastination and Reaction

Fig. 2. (a) Fit of the continuous model for courses with more than 500 visits. (b) Course semester with different course deadlines, and simulated cox process intensity. Notice that we intent to reproduce the statistical behavior as opposed to a smooth fit.

C. *Procrastination and Reaction* PR

The exemplary course data in Fig. 1 shows a population's reaction to an examination deadline. The reaction manifests itself in terms of first increasing and then peaking activity in the time between January 2013 and February 2013. In this example, reactions are observable on a time scale of the order of weeks. Other courses however, were observed to elicit shorter reaction times in that students tried to cover most of the material in just a few days. This renders the problem of data driven prediction of user behavior into a rather considerable challenge, since only a few data points are available in the region of exited activity. Also, many statistical models do not capture the intrinsic non-stationary nature of the phenomenon where the awareness of an upcoming deadline causes changes in the activity.

In order to model the temporal evolution of the daily rate $V_d(t)$, we propose an ordinary differential equation that accounts for the following minimal principles [7], [8]:

1) users try to learn the whole material by the time the deadline approaches
2) there is a procrastination parameter β which establishes when the visits to the system become a priority
3) users react over time according to some function $g(t)$.

We will model the function $g(t)$ as a power of time. This accords with theories of human behavior dynamics which make similar assumptions, for instance, to model how attention to news items, memes and games declines over time [9]–[12]. In our model however, the inverse behavior is expected, attention rises as time progresses. Hence, we expect a positive power as $g(t) \propto t^{\alpha}$. We thus consider the following differential equation

$$\frac{d}{dt}V_d(t) = \left(\frac{\alpha}{\beta}\right)\left(\frac{t}{\beta}\right)^{\alpha}\Big[T_e - V_d(t)\Big] \tag{5}$$

which is akin to that of an epidemic in that students which engage with the platform are thought of as *infected*.

In contrast to real epidemics, however, infections are exclusively driven by time and not by other infected students and they are limited by the amount of material which can be covered daily. The role of the infection rate is played by the *reaction rate*

$$g(t) = \left(\frac{t}{\beta}\right)^{\alpha} \tag{6}$$

and the term α/β is included for scaling. Finally, $T_e - V(t)$ accounts for natural limitations of the number of visits, since T_e is the maximum number of visits expected per day and is limited by the number of users and the average number of visits the course material needs to be covered.

Note that (5) has a close form solution

$$V_d(t) = \begin{cases} T_e - (T_e - V_0)e^{-(\frac{t}{\beta})^{\alpha}} & \text{if } t < t_d \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Also note that if $t > \beta$ the fraction in the exponent is bigger than one and the expression $(t/\beta)^{\alpha}$ grows rapidly. The negative exponent causes the exponent term to die out and $V_d(t) \to T_e$. V_0 is the initial daily rate of visits $V_d(0) = V_0$. The steepness of this jump depends on the value of α/β. If we derive (7) and evaluate at β we obtain $\frac{\alpha(N-N_0)}{e\beta}$, i.e. the slope at the reaction date. The time t_d indicates the day of the examination after which activities typically drop suddenly as students do not need to study urgently anymore.

It is important to note that, as a continuous model, equation (7) holds for courses which have a large enough population of attendees and a large enough workload. A large workload will guarantee that the users will remain engaged over several days, as each user requires many visits in order to cover the material.

For the purpose of parameter estimation from data, we denote the complete set of parameters of our dynamical model as

$$\theta_{pr} = \{T_e, V_0, \beta, \alpha\} \tag{8}$$

and determine θ_{pr} as those parameters that minimizes the following error

$$D(t, \theta_{pr}) = \sum_{\tau=1}^{T_d} (V_d^d(\tau) - \hat{V}_d^d(\tau)[\theta_{pr}])^2 \quad (9)$$

To minimize this expression, we resort to the *Levenberg-Marquadt* algorithm and let $\tau = 1, ..., \tau_d$ vary over the whole number of days of the semester. The fluctuating nature of the distribution of visits is accounted for via a Gaussian noise assumption with variance σ_n^2.

D. Piecewise PR for Course Workload

Although some courses will have the characteristics required by the continuous model in (7), most of the behavior related to a course will be characterized by rapid shocks occurring on days which define a deadline of some sort (examination or course work). This behavior, however, can be easily incorporated into (7) by requiring a high value of α. This will produce a high visit rate on only a few days around β, up to the deadline.

To fully specify a semester we need then to define one such shock for each deadline and therefore assume

$$V_d(t) = T_a - (T_a - V_0)e^{-(\frac{t}{\tau_a - \delta})^\alpha} \quad (10)$$

if $t_{a-1} < t < t_a$ which provides a piecewise approximation of our model, i.e. one solution of (5) for each course deadline, where $t_0 = 0$ and t_a defines the point in time of deadline $a \in \{0, 1, ..., M\}$ where M is the number of deadlines in a course. For simplicity, we assume that all deadlines are separated by δ days from day at which students begin to react (reaction width). Finally, we let T_a be the number of visits of the population at that deadline. Since we do not know in advance how many visits will happen for a given deadline, we model shocks in terms of random variables. From our empirical data we found via Kolmogorov-Smirnoff tests that the distribution that best fits our model follows the gamma distribution

$$Gamma(T|\rho, \nu) = \frac{\rho^\nu}{\Gamma(\nu)} T^{\nu-1} e^{-\rho T} \quad (11)$$

This is again very much in line with known models of human attention [9].

Finally, we observe that this kind of assumption in which we define the intensity function of a inhomogeneous Poisson process trough another stochastic process is known as a doubly stochastic Poisson process or as a Cox process [13].

E. Cascade Rates

In earlier related work [5], it has been pointed out that short term behaviors of user populations can be modeled via a Poisson process in which another rate is imposed after the initial visit. Since we do not know which of the visits will generate a cascade of activities, we define a variable p_c as the proportion of initial Poisson events which give rise to a cascade. Furthermore, λ_c defines the rate of the uniform Poisson process which characterizes the Poisson distribution.

Algorithm 1 Generating a Cox Process

Require: Semester τ_s, Distribution Parameters θ_{pw}, E
1: $\{\hat{t}_i\}_{i=1}^E \sim \text{Uniform}(\tau_s)$
2: $T_a \sim P(T|\theta)$
3: $U \sim \emptyset$
4: **for** $i \leftarrow 1, ..., E$ **do**
5: $u_i \sim \text{Uniform}(0, 1)$
6: $r_i \sim V(t_i|T_a)$
7: **if** $u_i < r_i$ **then**
8: $U \leftarrow U \cup \hat{t}_i$
9: **end if**
10: **end for**
11: **return** U

F. PPRC

Summarizing all of the above, we refer to our full model as the piecewise, procrastination reaction model (PPRC). The complete set of parameters of our model is given by

$$\theta_{pw} = \left\{\delta, V_0, \sigma_0, \nu, \rho, P_d(t), \lambda_c, p_c\right\} \quad (12)$$

and characterizes the main aspects of human behavior for the inter-event time distribution. Circadian characteristic are captured by $P_d(t)$, the reaction of the population towards a given task is parametrized by ν, ρ, and δ, baseline behaviors are expressed via V_0 and σ_0 and short term behaviors via λ_c and p_c.

III. Methodology

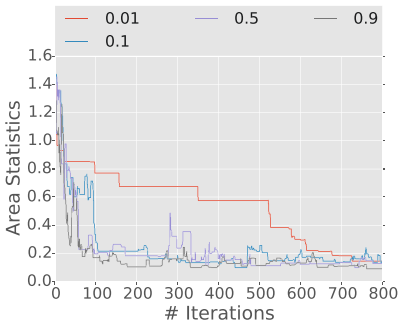
Next, we outline the procedure we use for model fitting. First we show how to obtain the procrastination reaction parameters and then introduce an algorithm for simulating a Cox process. Finally, we discuss a training procedure based on simulated annealing which allows us to obtain the parameters for the PPRC θ_{pw} as defined by (12).

A. Thinning Algorithm

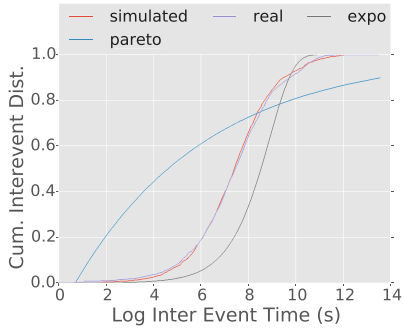
In order to simulate and generate data from our model, we proceed via a modification of the rejection sample algorithm for point data, known as thinning.

Given our overall collection of observed data, we intent to generate a set of seconds $\{t_i\}$ ranging from 0 to τ_s the total number of seconds in one semester. Traditionally, inhomogeneous Poisson process generation [14], [15] requires us to sample from a uniform Poisson distribution via a maximum intensity λ^*, since an inhomogeneous Poisson process with intensity $\lambda(s)$ requires that its number of events to be distributed via $N(\mathcal{S}) = \int \lambda(s)ds$.

In our case however, we do not know the contribution to the distribution of the number of events in a cascade so we directly generate E events from a uniform distribution over the interval $(0, \tau_s)$. We then generate a gamma distributed sample T_a (see again (11)) and random noise from $\mathcal{N}(0, \sigma_n)$. This then allows us to create the stochastic intensity function of the PPRC model.



(a) Area test statistic obtained from iterating a simulated annealing procedure for different temperature values.



(b) Logarithm of the cumulative distribution of the inter-event times

Fig. 3. Exemplary results of the behavior of model in training and empirical data fitting.

We generate the desired intensity shape by using a set of uniform random variates $\{u_i\}_{i=1}^{E}$ in $[0,1]$ and evaluate each t_i in the obtained PPRC function. This way (i.e. using algorithm 1 for Cox process generation), we obtain a number $\tilde{E}$ of events $\{t_j\}_j^{\tilde{E}} = 1$. Finally we incorporate short term behaviors by choosing $f_c = p_c\tilde{E}$ different events. From these events, we generate a sample from a Poisson process with rate λ_c.

B. Training

Given a sample of empirical point visits $\{t_e\}$, we next wish to estimate the parameters of our model θ_{pw} which best reproduce the data. The daily behavior of the model as established by $P_d(t)$ can be obtained directly from the histogram of the hours of $\{t_e\}$. To obtain the PPRC parameters we first need to obtain the daily visits by a histogram of the point data for each day.

We then obtain the peak values corresponding to the relevant course work by normalizing the daily visits histogram and consecutively choosing the biggest peaks of the distribution (located at $\{t_a\}$ with values $\{T_a\}$) until the cumulative distribution posses a standard deviation bigger than 0.25, up to a maximum of 24 different peaks (which would correspond to the 24 weeks per semester and a maximum of one homework per week).

We obtain the parameters of the gamma distribution using maximum likelihood estimation from the values of the $\{T_a\}$ found by this procedure. In order to train the remaining parameters of our model, we require that the inter-event distribution $P_{\mathcal{M}}(u|\theta_{pw})$ as sampled via the numerical Cox algorithm reproduces the distribution of the empirical cumulative distribution $P_{\mathcal{D}}(u)$. The objective function, we consider for for maximization s given by the area test statistic $A = \int |P_{\mathcal{D}}(u) - P_{\mathcal{M}}(u|\theta_{pw})|du$ where we choose $u = \log(t)$ for numerical convenience. We thus obtain the cumulative model distribution as a sample statistic form our model. As such, for a given values of the parameters different samples will generate different values of the area statistic. In order to minimize the stochastic surface defined by the parameters, we use simulated annealing [16] by random displacement on the space defined by the variables $(V_0, \sigma_0, \delta, p_c, \lambda_c, E)$. As an example of the results of this model fitting procedure, we show the outcome w.r.t. a computer science course in Fig. 3 where we compare the performance our model to a Pareto- and exponential distribution chosen as baseline models.

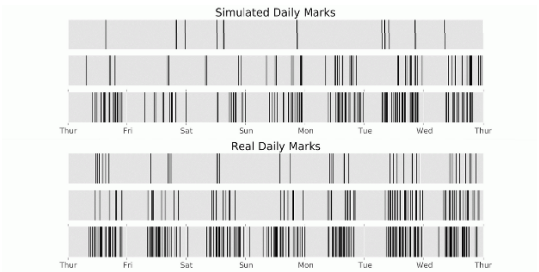


Fig. 4. Real distribution of visits over a period three weeks before an examination deadline related to one of the courses in our data set (lower panel), and simulated point process of visits using the Cox process discussed in the text. Note that idle periods reflect reduced activities over night.

IV. Results

In a series of practical experiments, we trained our models for time marks of 20 different courses. In each case, we initialized the model parameters to $E = 4000$, $\lambda_r = 1500$ $p_c = 0.7$ $V_0 = 5$ $\sigma_0 = 1.$ $\delta = 0.1$ and used a total of 4000 iterations and an annealing temperature of 0.2. After training, we obtained a average Kolmogorov-Smirnoff (KS) divergence statistic of 0.03 ± 0.01 for the whole data set as well as a cascade rate of cascade $1930 \pm 691[s]$ for an average cascade rate of 30 minutes. Finally, the reaction width δ found in our data was 2.32 ± 1.6 days before the deadline.

Table IV presents goodness-of-fit statistics for a random selection of highly attended courses. Overall, the Pairwise Procrastination Reaction Cascade model proposed in this paper was found to fit (almost surprisingly) well to our empirical

TABLE I
AREA- AND KOLMOGOROV-SMIRNOFF STATISTICS FOR RANDOMLY SELECTED COURSES IN OUR DATA SET.

Course Name	Area Statistic	KS Divergence
Computer Science	0.038	0.016
Databases	0.053	0.014
U.S.-American Literature	0.054	0.019
School Studies	0.13	0.051
Multivariable Calculus	0.075	0.026

data. This suggests that our proposed extension of previous inhomogeneous Poisson processes models of human activity dynamics on the Web [5] does indeed capture the peculiar dynamics observed on eLearnig sites.

V. CONCLUSIONS

Our goal in this paper was to devise a model for the access behavior of a population of students on a university eLearning site. The particular challenge was to account for characteristic procrastination and reaction patterns observable prior to final examination deadlines.

Though rather intricate machine learning techniques are necessary to fit our model to the given data, we emphasize that the model itself is not a black box but was derived from first principles. This is to say that each of the constituent parts of our model are interpretable. In particular, they represent characteristics of human behavior, circadian cycles, the procrastination and reaction in the present of a deadline, as well as short term use of the system. These components were integrated via a dynamical Cox process model which is intrinsically non stationary and allows for analyzing data data of poor quality (few observations only). In practical experiments, the resulting Pairwise Procrastination Reaction Cascade model was found to be well capable of reproducing the statistics of the behavior of a student population.

To the best of our knowledge, the work reported here is the first such study undertaken on a large data set of access patterns to an eLearning platform. Accordingly, there are numerous direction for future work. In particular, we are currently working on extending our approach towards a practical application that allows teachers to schedule their deadlines for course work and final examinations such that the expected workload for students is more equally distributed over course of a semester.

REFERENCES

[1] A. L. Barabasi, "The origin of bursts and heavy tails in human dynamics," *Nature*, vol. 435, no. 7039, pp. 207–211, 2005.
[2] F. Wu and B. Huberman, "Novelty and collective attention," *PNAS*, vol. 104, no. 45, pp. 17 599–17 601, 2007.
[3] R. Sifa, F. Hadiji, J. Runge, A. Drachen, K. Kersting, and C. Bauckhage, "Predicting Purchase Decisions in Mobile Free-to-Play Games," in *Proc. of AAAI AIIDE*, 2015.
[4] C. Ojeda, K. Cvejoski, R. Sifa, and C. Bauckhage, "Variable Attention and Variable Noise: Forecasting User Activity," in *Proc. of LWDA KDML*, 2016.
[5] R. D. Malmgren, D. B. Stouffer, A. E. Motter, and L. A. N. Amaral, "A Poissonian explanation for heavy tails in e-mail communication," *PNAS*, vol. 105, no. 47, pp. 18 153–18 158, 2008. [Online]. Available: http://www.pnas.org/content/105/47/18153.abstract
[6] R. G. Gallager and R. G. Gallager, *Discrete stochastic processes*. Kluwer Academic Publishers Boston, 1996, vol. 101.
[7] M. A. Alvarez, D. Luengo, and N. D. Lawrence, "Latent force models," in *Proc. AISTATS*, 2009.
[8] T. Gunter, C. Lloyd, M. A. Osborne, and S. J. Roberts, "Efficient bayesian nonparametric modelling of structured point processes," *arXiv preprint arXiv:1407.6949*, 2014.
[9] C. Bauckhage, "Insights into internet memes." in *Proc. ICWSM*, 2011.
[10] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos, "Rise and fall patterns of information diffusion: model and implications," in *Proc. KDD*, 2012.
[11] C. Bauckhage, K. Kersting, and F. Hadiji, "Mathematical models of fads explain the temporal dynamics of internet memes," in *Proc. ICWSM*, 2013.
[12] R. Sifa, C. Bauckhage, and A. Drachen, "The Playtime Principle: Large-scale Cross-games Interest Modeling," in *Proc. of IEEE CIG*, 2014.
[13] D. R. Cox, "Some statistical methods connected with series of events," *J. Royal Statistical Society B*, vol. 17, no. 2, pp. 129–164, 1955.
[14] P. A. Lewis and G. S. Shedler, "Simulation of nonhomogeneous poisson processes by thinning," *Naval Research Logistics Quarterly*, vol. 26, no. 3, pp. 403–413, 1979.
[15] R. P. Adams, I. Murray, and D. J. MacKay, "Tractable nonparametric bayesian inference in poisson processes with gaussian process intensities," in *Proc. ICML*, 2009.
[16] S. Kirkpatrick, M. Vecchi *et al.*, "Optimization by simmulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

Investigating and Forecasting User Activities in Newsblogs: A Study of Seasonality, Volatility and Attention Burst

Christian Bauckhage, César Ojeda and Rafet Sifa
Fraunhofer IAIS
St. Augustin, Germany

Christian Backhage and Rafet Sifa
University of Bonn
Bonn, Germany

***Abstract*—The study of collective attention is a major topic in the area of Web science as we are interested to know how a particular news topic or meme is gaining or losing popularity over time. Recent research focused on developing methods which quantify the success and popularity of topics and studyied their dynamics over time. Yet, the aggregate behavior of users across content creation platforms has been largely ignored even though the popularity of news items is also linked to the way users interact with the Web platforms. In this paper, we present a novel framework of research which studies the shift of attentions of population over newsblogs. We concentrate on the commenting behavior of users for news articles which serves as a proxy for attention to Web content. We make use of methods from signal processing and econometrics to uncover patterns in the behavior of users which then allow us to simulate and hence to forecast the behavior of a population once an attention shift occurs. Studying a data set of over 200 blogs with 14 million news posts, we found periodic regularities in the commenting behavior. Namely, cycles of 7 days as well as 24 days of activity which may be related to known scales of meme lifetimes.**

I. Introduction

Much recent research on Web analytics has concentrated on developing theories of collective attention where the main object of study is the evolution of the popularity of topics, ideas, or sets of news [1], [2]. In this context, the concept of a *meme* has arisen as the main *atom* of modern quantitative social science [1], [3] and researchers seek to understand whether a particular meme will remain popular and, if so, for how long. Under this line of research, virality is the main phenomenon to model. Usually, a contagious (i.e. network based) approach is followed, and virality is literally treated as an infection in a given population: as an item becomes popular, i.e. as a piece of news is retweeted or discussed in the media, the population unit, whether blog or twitter account, is considered infected.

It is important to note that, this paradigm of research, ignores the impact of the source of the meme. The website or content generating media plays a key role in generating (or hindering) the evolution of a particular idea or topic. Naturally, if a given website has a large number of users, it is likely, that certain topics become popular and capture the attention of the population. We might thus ask whether the baseline behavior of the users of a website is enough to generate popularity. Is it the content which is popular or is it the website which is popular? If we can use the baseline behavior of a given website as a reference, we might use such information as a general guideline for forecasting. Information regarding the population behavior over a website will provide the basis for understanding both the evolution of that particular website as well as the possible success of the future content.

The population behavior on a given website is unavoidably stochastic, as we cannot know a priori whether a particular user or a set of users will visit or not (see Fig. 1 for an examplary time series of activities of blog commenters). Statistical time series analysis has a rich history of success in fields as diverse as electronics, computer science, and economics and we know that, given that relevant information regarding the behavior of the system is properly modeled in the phenomena that is measured, and randomness is realized under proper bounds, estimates can be achieved and predictive analytics becomes possible.

In this work we present a bottom up case study to analyze attention of blog users (particulary commenters) to understand and predict their activity patterns which exploits the fact that many websites as well as blogs have years of user history which can be mined in search for relevant patterns.

II. Related Work

The study of information diffusion on the Web intents to model pathways and dynamics trough which ideas propagates. One of the goals is to infer the structure of networks in which information propagates [4]. Researchers often try to devise equations which govern the aggregate behavior [2]; these equations typically have parameters which depend on the population size, the rate of the spreading process, and universal features of how attention fades. They model time series which show how much activity a particular topic or meme attract. Forecasting can be then performed once the parameters of a particular population have been determined. In [5] Matsubara et al. learned the parameters of the attention time series related to the first Harry Potter movie and then predicted the behavior for the next movies by measuring only the initial population reaction. More importantly, the *natural* limitations of human attention and human behavior have shown to define the overall behavior of the population as they access the information [6], [7], [8], [9].

Blog dynamics are traditionally studied in the context of conversation trees established between different blogs [10],

P. Haber et al. (Hrsg.), *Data Science – Analytics and Applications*, http://doi.org/10.1007/978-3-658-19287-7_2

number of blogs	203
number of posts	713,122
number of comments	14,883,752
time span of activity	2006-2014

TABLE I: Characteristics of the Wordpress Data Set

[11]. In this context, model of the structural and dynamical properties of networks of hyperlinks are sought for. McGlohon et al. [11] try to find representative temporal and topological features for grouping blogs together and to understand how information propagates among different blogs. The model in [10] defines rules which indicate how the different links are created. Typical properties include the degree distribution of the edges to a blog, which presents a power law behavior, the size of conversation trees (how many posts are created in a given theme) and the popularity in time, as measured with the amount of edges in time.

Here, we intend to make use of methods of statistical time series modeling in order to study and forecast the behavior of a population on a given webpage. In particular, we focus on time series of users commenting patterns. We base our work on well establish methods in economics [12], [13] and signal processing [14]. In a nutshell, we attempt to describe the statistical process which generates the time series, modeling both temporal correlations as well as signal noise. Due to the world financial crises of the years 2008 a great deal of attention in the field is devoted at understanding volatility or strong fluctuations in the market [15], [16]. At their core, the problems of volatility and collective attention shifts are similar and we exploit these similarities in our approach.

III. Data Set Description

The empirical basis for our work consists in a collection of blogs hosted on *Wordpress*[1]. Wordpress blogs are personal or company owned web sites that allow users to create content in the form of posts. Users (or, in blogger terminology, writers or authors) decide whether or not their readers can comment on particular posts. Comments on posts provide a significant proxy when it comes to measuring the attention a topic receives.

Wordpress is a content management system and a frequently used blogging platform accounting for more than 23.3% of the top 10 million websites as of January 2015[2]. As of this writing, there are 43 million posts and more than 60 millions users. Notably, the user-base of Wordpress ranges from individual hobby bloggers to publishers such as Time magazine[3]. And while traditional mass media limit the number of topics accessible to the public [17], blogs have arisen as a *social medium* allowing for personal opinion and information variety. Nevertheless, inequality is a pervasive characteristic of social phenomena [18], [19] and many of the consumers of blog contents concentrate on sites with a large community of frequent users.

Blog URL	News Genre	Viral Post Count
le-grove.co.uk	Sport	620
politicalticker.blogs.cnn.com	Politics	542
order-order.com	Politics	294
sloone.wordpress.com	Personal	248
technologizer.com	Technology	136
snsdkorean.wordpress.com	Entertainment	116
collegecandy.com	Magazine	108
seokyualways.wordpress.com	Personal	96
religion.blogs.cnn.com	Religion	93
kickdefella.wordpress.com	Personal	89

TABLE II: Top 10 Newsblogs from our dataset.

Our dataset contains the entire post and comment history of newsblogs (from year 2006 to 2014) that ever made it to the daily updated list of *Blogs of the Day*[4]. Having crawled the content of over 6580 different blogs, we choose to work with the time series of activities through comments and posting of the most active (throughout the available time span) 203 blogs from variety of genres in order to study long term dynamics. The blogs we analyze overall contain 713,122 posts and 14,883,752 comments. Table I summarizes the overall characteristics of our data and Table II shows the top 10 blogs sorted with respect to the number of viral posts. So as to protect the identity of commenters, we randomly hashed the author identities and key features we extracted from each post are

- CommentsTime: Time of each comment
- MainURL: URL of the post
- CommentsInfo: Content of the comment
- Title: Title of the post
- PubDate: Date of last update of the post
- Blog: URL of the blog
- Info: Summary of the Post

IV. Empirical Observations

Our main objective in this paper is understanding the dynamics of the aggregate behavior of a population of users of a webpage or blog. For each blog we determine the number of comments C_t on all posts at time t. Figure 1 presents a typical example of such a time series of daily commenting behavior spanning a period of 3 years together with a monthly moving average. Averaged over months, the time series shows a slowly varying behavior which reflects the fact that, over time, the attention the blog receives does not vary much. On the other hand, on the time scale of days rapid variations can be observed, as well as strong peaks which indicate the presence of particularly engaging content. We aim at applying forecasting methods to these very fluctuations. The main goal of our work is to propose a forecasting procedure that can characterize these shifts of attention.

We assume that a prototypical webpage has a steady user base. This user base can be understood as a *set of followers* and as different followers visit the website, comments will appear as natural stochastic fluctuations. However, if a particular news item is highly relevant or *hot* for a segment of the population,

[1] https://www.wordpress.com/

[2] Information retrieved on 5.3.2013 from http://w3techs.com/technologies/overview/content_management/all/

[3] https://vip.wordpress.com/2014/03/06/time-com-launches-on-wordpress-com-vip/

[4] https://botd.wordpress.com/

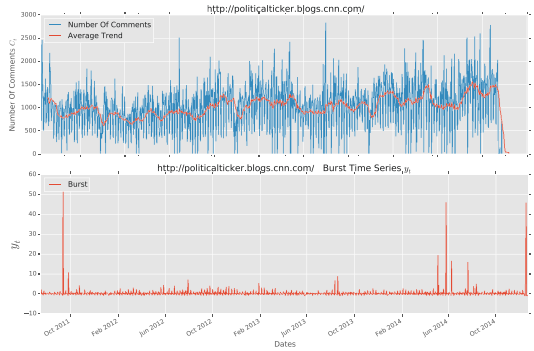


Fig. 1: Time series of the number of comments C_t and bursts y_t for a newsblog. The upper figure shows the daily fluctuations of activities (blue) and the monthly moving average trend (red). The lower figure shows a burst representation that allows for locating noticeable changes in the overall behavior.

then users will show more activity, increase their commenting rate, and also share the content with others users *which are not part of the baseline set of followers of the blog*. Consequently, some of the comments will appear as a result of a spreading process due to the web site followers. *The popularity of web page translate into popularity of the content*.

Assuming a somewhat stable set of users has the added advantage of exploiting seasonality effects, as people are known to follow seasonal behavior [20]. Such periodicities can help to better understand the repeating behavior (for instance weekly visits of average users) as well as to forecast activities.

V. Time Series Analysis

We model the user commenting behavior as a discrete stochastic process, i.e. as a collection of random variables $X_1, X_2, X_3, ..., X_k$ indexed by time. Since we aim at attention modeling, our first variable of study is C_t, the number of comments an entire blog has at time t. Considering daily samples accords with the pace of publication in most blogs.

A. Seasonality

One approach to time series prediction using stochastic processes requires detecting periodicities. If are recovered by detecting their frequencies and associated amplitudes, we can predict the behavior at each cycle. A common method considers power spectra of the stochastic process at hand via the discrete Fourier transform

$$f_x(\omega) = \sqrt{\frac{1}{T}} \sum_{t=1}^{T} x_t \exp -i\omega t. \tag{1}$$

We then square the transform

$$I(\omega) = \frac{1}{2\pi} f_x(\omega) \hat{f}_x(\omega) \tag{2}$$

to obtain the so called periodogram of a time series.

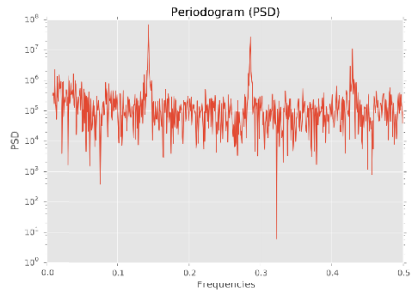


Fig. 2: Seasonality study through a semilog plot of the periodogram for estimation of the power spectral density from an example newsblog. Notice the presence of 3 frequencies among the noisy behavior that account for commenting activity in period of three-and-a-half and seven days.

If seasonal behavior is present, peaks will appear in the periodigram. For example, Fig.2 shows the periodogram for a time series of comment counts from our dataset. Notice the peaks at frequencies of 0.13, 0.29 and 0.31 which account for periods in the comment patterns of *weekly* and *half weekly* seasonal behavior indicating that the users of that particular newsblog leave comments in three-and-a-half and seven days frequencies.

B. Burst and Volatility

In order to characterize the popularity of a particular news source, we require a quantitative measurement of attention shifts. These can described via an iter-burst measure [21], namely

$$y_t = \frac{C_t - C_{t-1}}{C_t}. \tag{3}$$

Eq. (3) is also know as the logarithmic derivative of the process C_t. Due to its rescaled nature, y_t allows for comparison between different newsblogs and times, but, for our data set, it provides limited value for forecasting. We thus also consider the detrended comment count $\tilde{C}_t$ which is obtained after using the *Baxter-King* filter (see the Appendix).

C. Modeling Volatility

Having introduced $\tilde{C}_t$, we next need an approach that can account for fluctuations. Since after the filter is applied we obtain a zero mean behavior, we can assume that the comments behavior will vary as

$$\tilde{C}_t = \sigma_t \epsilon_t \tag{4}$$

where σ_t represents the standard deviation of the fluctuation at time t and $\epsilon_t \sim \mathcal{N}(0,1)$ a noise value at time t, sampled from a normal distribution of mean 0 and variance 1. In finance, when modeling the fluctuations in returns (as opposed to $\tilde{C}_t$) σ_t is known as the volatility. The simplest model for volatility occur in econometrics under the names of $ARCH$ (autoregressive conditional heteroscedasticity) and generalized $ARCH$ or

$GARCH$ [22]. Here, heteroscedasticity refers to variations in the fluctuation of the variable of interest, in our case C_t. Formally, this implies that the covariance $Cov(C_t, C_{t-k})$ depends on time t. The dependence is modelled trough past values of the noise e_t as well as past values of σ_t

$$\sigma_t^2 = \omega + \sum_{k=1}^{q} \alpha_k \epsilon_{t-k}^2 + \sum_{l=1}^{q} \beta_l \sigma_{t-l}^2. \quad (5)$$

The values of p and q determine how correlated σ_t is with past fluctuations so that the model is called $GARCH(p, q)$. To learn these parameters from data, we have to take into account that for $GARCH(1, 0)$ the square of the fluctuations are $\tilde{C}_t^2 = \alpha_0 + \alpha_1 \tilde{C}_{t-1}^2 + error$, i.e. that they behave as an $AR(1)$ (an autoregressive model of order 1) [12]. Hence, the values of the partial autocorrelations of the square variable give us a hint of the dependence of the model.

VI. Main Results

A. Attention Proxy

Traditional attention proxies for collective attention comprise direct measures such as the number of hyperlinks to a web site, retweets, or the number of likes via the facebook platform. Sometimes an internal measure of the website is preferred, for instance, the web site *digg.com* uses a user based ranking mechanism which ultimately decides which content is displayed on the main page. In order to validate our use of comments as an attention proxy, we make use of the fact that Wordpress regularly features the most popular posts[5]. In the following, we refer to normal posts as posts not presented in this list, whereas viral or important posts as the posts presented in the list and we note that, indeed, posts features on the most popular list attract more comments than others.

Since we work with a dataset that covers several years of activity, it is important to note that the amount of comments expected in a given year is different from the amount of comments in another if the blog shows growth or decay in its user base. Yet, this issue is accounted for by our use of *de-trended behavior* $\tilde{C}_t$.

For our whole data set of relevant blogs, we observe average fluctuation value for the important posts of 71.12 whereas for the normal blogs an average fluctuation value of -96.14. This shows that *most of the comment behavior arises from popular or viral posts*.

Figure 3 shows the distribution of comments for normal and important posts where we observe positive skew distributions that can be modeled using pareto and lognorm distributions (see the Appendix). The pareto behavior indicates that there are strong fluctuations in the normal data set, meaning that there are in fact important posts not considered by the Wordpress listing procedure.

B. Seasonality

In this section we perform the periodogram analysis to identify the cycles of commentors leaving comments in the

[5]https://botd.wordpress.com/top-posts/

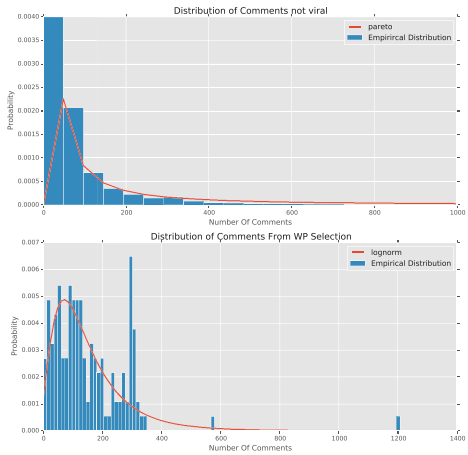


Fig. 3: The distribution of comments. The upper figure shows the distribution of comments for posts pertaining to the non-relevant data set. Pareto distribution is fitted with 43 average value and p value = 0.43. The lower figure on the other hand shows the distribution of comments for relevant posts as determined from the Wordpress data set average 254 comments per post and p value = 0.73.

newsblogs. It is important to note that such an analysis should be realized for each blog independently, as it will reveal the cycles of the followers or induced cycles due to the posting patterns of the particular website we are analyzing. Nevertheless, it is commonly known that both human behavior and virality patterns often show universal trends. For instance, attention to memes grows and fades over time in highly regular patterns [1]. Also, people show repetitive behavior because of work schedules and life habits [20]. If there is in fact such universal behavior for our case, a periodogram analysis should reveal common characteristics across different blogs.

Indeed, based on periodogram analysis, we obtained the most relevant frequencies for each periodogram as the frequencies above 4 standard deviations over the statistics of each individually calculated periodogram. Figure 4 shows the histogram of the relevant frequencies where we observe relevant peaks at frequencies of *17 and 7 days*.

The peak with the largest frequency value of 7 days can be attributed to a weekly behavior of the users. Business days constrain the publishing efforts of websites as well as the visiting patterns of users. The 17-day period on the other hand, seems to correspond to the natural attention scale of publishing behavior. A news content provider for example, publishes several posts on a given topic and this 17 day frequency might correspond to the saturation of attention resourses people can

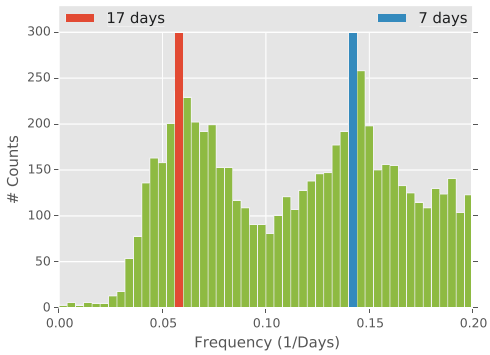


Fig. 4: Seasonality study through analyzing the histogram for the relevant frequencies over the whole population of blogs. For each blog we selected the frequencies which exceed the 4 standard deviation. We observe peaks for 7 and 17 days indicating users' periodic commenting frequencies.

allocate to different subjects [9]. Unless we perform an actual content analysis, however, we will not be able to pinpoint the true latent cause of this periodicity. Neverthelss, our results show a consistent appearance of this frequency throughout our Wordpress data set.

C. Forecasting

Having analyzed the nature of user behavior and havinf uncovered periodic patterns in the behavior of commentors, we next approach the ultimate goal of analytics which, in our case, is forecasting user fluctuations.

1) A Forecasting Protocol: To this end, We propose the following protocol in order to study the fluctuations in attention to blogs:

- Create the time series of comments per day C_t (different time scales such as hours, days, or weeks may be considered, provided that enough samples are available).
- Apply a detrending procedure such as the Baxter King filter in order to obtain a stationary version of the given time series.
- Obtain the peridogram of the time series so as to identify relevant frequencies.
- Compute the auto correlation (Eq. (9)) and the partial autocorrelation of $\tilde{C}_t$.
- If **relevant** autocorrelations can be observed, the blog is amenable for forecasting of fluctuations.
- Fit the $ARCH$ or $GARCH$ model to the time series.
- Once the values of β_p and α_q are known, we can obtain the conditional volatility, i.e. the values of σ_t given past values of q and p.

In order to fit the parameters of the $ARCH$ and $GARCH$ processes, many commercial and free software packages [6] are available [22]. We performed a statistical test of heterocedasticity for all of the blogs in our data set [16] and selected the top 100 web sites. We present the results for the top 10 blogs in Table III. We fited $GARCH$ models with a grid search on the paramters $p \in \{1, ..., 10\}$ and $q \in \{1, ..., 10\}$ and selected the best model according to Akaike Information Criterion (AIC) [23]. For comparison, we obtain the realized volatility [22] i.e. the standard deviation for the past 14 days window (σ for $\{\tilde{C}_{i-14}, ..., \tilde{C}_{i-1}, \tilde{C}_i\}$) and compared to the conditional volatility σ_t as predicted by $GARCH$ given the past values of the models one day in advance (according to the selected q and p). Figure 5 shows an examplary $GARCH$ fit of the fluctuations of the commenting behavior of a newsblog. For the top 100 blogs, we obtained average MSE values of 0.739100 for the models with best AIC values and 0.663600 for the models with p=1,q=1.

[6] An example open source python package for fitting ARCH models can be found under https://pypi.python.org/pypi/arch.

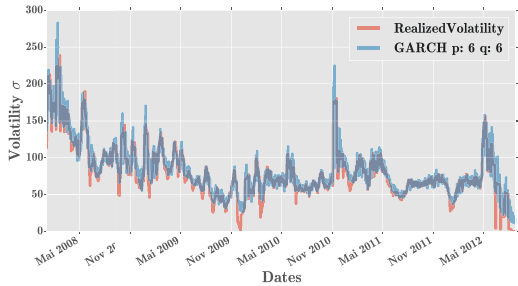


Fig. 5: Realized volatility vs Conditional Volatility as obtained from the best GARCH model for the http://marisacat.wordpress.com/ website.

Blogs	Best Model	MSE	MSE p=1 q=1	# Comments
le-grove.co.uk/	(7, 7)	2.00	1.87	1625633
politicalticker.blogs.cnn.com/	(3, 2)	4.48	4.16	1461266
order-order.com/	(5, 5)	4.87	3.80	1289484
sloone.wordpress.com/	(8, 7)	1.39	1.16	93493
technologizer.com/	(9, 1)	594.34	649.79	165856
snsdkorean.wordpress.com/	(2, 5)	0.40	0.28	83081
collegecandy.com/	(9, 1)	0.71	0.61	51495
seokyualways.wordpress.com/	(1, 6)	3.76	2.41	130460
religion.blogs.cnn.com/	(9, 9)	21.66	18.74	2426437
kickdefella.wordpress.com/	(6, 8)	0.22	0.17	39312

TABLE III: Fitting results for the top 10 newsblogs.

VII. Conclusions and Future Work

We were concerned with the problem of understanding the dynamics of the collective attention of populations of readers of newsblogs and presented a novel time series analysis approach. Our approach relies on two main concepts: seasonality and volatility. The seasonality is uncovered by the analysis of the periodogram function of the time series of comments. Volatility, on the other hand, is uncovered by analyzing the deviation of the variable fluctuations in time. We found universal seasonal behavior of newsblogs users in our dataset which hint ate universal patterns of human behavior. Additionally, we showed how $GARCH$ modeling of the time series presents a way to forecast possible future scenarios for

users of newsblogs as we can generate fluctuations of user attention based on the previous history. Such patterns can be exploited to perform informed decision making, to develop new time based marketing and advertisement strategies, and to present content to new as well as loyal users.

In future work, we will focus on developing better attention and credibility metrics [24] since cyclic behavior might generate fictitious attention shifts. It is important to note that the periodogram uncovers the importance and period of the cycles present, but in order to fully recover the information at hand, a detailed analysis of the population is yet to be performed. There is also a need to uncover the meaning of the stochastic process parameters from first principles, i.e. from the individual users' behavior. As opposed to finance, where only time series data is available, news websites offer the possibility of tracking individual user behavior over time. Hence, since we might have more information about the individuals, a more detailed study of volatility could provide even better attention models.

REFERENCES

[1] C. Bauckhage, "Insights into internet memes." in *Proc. of ICWSM*, 2011.
[2] A. Vespignani, "Modelling dynamical processes in complex socio-technical systems," *Nature Physics*, vol. 8, no. 1, 2012.
[3] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *Proc. of ACM SIGKDD*, 2009.
[4] J. Yang and J. Leskovec, "Modeling information diffusion in implicit networks," in *Proc. of ICDM*, 2010.
[5] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos, "Rise and fall patterns of information diffusion: model and implications," in *Proc. of ACM SIGKDD*, 2012.
[6] A.-L. Barabasi, "The origin of bursts and heavy tails in human dynamics," *Nature*, May 2005.
[7] R. Sifa, C. Bauckhage, and A. Drachen, "The Playtime Principle: Large-scale Cross-games Interest Modeling," in *Proc. of IEEE CIG*, 2014, pp. 365–373.
[8] F. Wu and B. A. Huberman, "Novelty and collective attention," *Proceedings of the National Academy of Sciences*, vol. 104, no. 45, 2007.
[9] L. Weng, A. Flammini, A. Vespignani, and F. Menczer, "Competition among memes in a world with limited attention," *Scientific reports*, vol. 2, 2012.
[10] M. Goetz, J. Leskovec, M. McGlohon, and C. Faloutsos, "Modeling blog dynamics." in *Proc. of ICWSM*, 2009.
[11] M. McGlohon, J. Leskovec, C. Faloutsos, M. Hurst, and N. Glance, "Finding patterns in blog shapes and blog evolution," in *Proc. of ICWSM*, 2007.
[12] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *Journal of the American statistical association*, vol. 74, no. 366a, 1979.
[13] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation," *Econometrica: Journal of the Econometric Society*, 1982.
[14] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, *Statistical and adaptive signal processing: spectral estimation, signal modeling, adaptive filtering, and array processing*. Artech House Norwood, 2005, vol. 46.
[15] C. T. Brownlees, R. F. Engle, and B. T. Kelly, "A practical guide to volatility forecasting through calm and storm," *Available at SSRN 1502915*, 2011.
[16] A. J. Patton and K. Sheppard, "Evaluating volatility and correlation forecasts," in *Handbook of financial time series*. Springer, 2009, pp. 801–838.
[17] E. S. Herman and N. Chomsky, *Manufacturing consent: The political economy of the mass media*. Random House, 2008.
[18] M. Newman, "Power laws, pareto distributions and zipf's law," *Contemporary Physics*, vol. 46, no. 5, 2005.
[19] G. Szabo and B. Huberman, "Predicting the Popularity of Online Content," *Comm. of the ACM*, vol. 53, no. 8, 2010.
[20] R. D. Malmgren, D. B. Stouffer, A. E. Motter, and L. A. Amaral, "A poissonian explanation for heavy tails in e-mail communication," 2008.
[21] J. Ratkiewicz, S. Fortunato, A. Flammini, F. Menczer, and A. Vespignani, "Characterizing and modeling the dynamics of online popularity," *Phys. Rev. Lett.*, vol. 105, Oct 2010.
[22] J. D. Hamilton, *Time series analysis*. Princeton university press Princeton, 1994, vol. 2.
[23] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.
[24] B. Ulicny and K. Baclawski, "New metrics for newsblog credibility," in *Proc. of ICWSM*, 2007.
[25] R. G. King and S. T. Rebelo, "Low frequency filtering and real business cycles," *Journal of Economic dynamics and Control*, vol. 17, no. 1, 1993.

VIII. APPENDIX

A. Distributions

1) Pareto Distribution:: Given the shape parameter α the probability density function (PDF) of the pareto distribution is assigned non-zero values starting from the scale parameter x_{min}. The PDF of pareto distribution is :

$$f_X(x;\alpha,x_{min}) = (\alpha-1)x_{min}^{\alpha-1}x^{-\alpha}. \tag{6}$$

2) Log-normal distribution:: A random variable whose logarithm is normally distributed follows lognormal distribution. Given respectively scale and location parameters σ and μ the PDF of the lognormal distribution is defined as:

$$f_X(x;\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}}\, e^{-\frac{(\ln x-\mu)^2}{2\sigma^2}}, \quad x>0. \tag{7}$$

B. Filters

The original role of the Baxter King filter [25] is to isolate the business cycle over a particular econometric time series. It is accomplished via a transformation of the stochastic process as $B(l)C_t = \tilde{C}_t$. The desired effect is accomplished by looking at how the transformation is carried in the spectral density. In the frequency space the filter is expressed via: $B^*(e^{i\omega}) = \sum_{\infty}^{\infty} a_j e^{i\omega j}$. This is impossible to perform since we only have a finite time series. We approximate via:

$$a_j = \begin{cases} a_j * +\theta & \text{if } |j|<J \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

We impose $B(1)=0$ eliminating zero frequencies and in turn rendering the process stationary.

1) Autocorrelations: The autocorrelation function measures the correlation of the values of the stochastic process at two different points in time. How similar two values are. In the case of signal S_t of period τ for example, the correlation will be maximum at the period. It is a function of the time lag between two points in the time series. It is defined as:

$$R(s,t) = \frac{E[(X_t-\mu)(X_s-\mu)]}{\sigma_t\sigma_s} \tag{9}$$

For a stochastic process x_t we define the partial correlation (the conditional correlation) is defined as

$$P(k) = \frac{Cov(x_t, x_{t-k})}{\sqrt{V_t V_k}} \tag{10}$$

where V_t and V_k are defined as

$$V_t = Var(x_t|x_{t-1},..,x_{t-k}) \tag{11}$$

$$V_k = Var(x_{t-k}|x_{t-1},..,x_{t-k-1}). \tag{12}$$

Knowledge-Based Short-Term Load-Forecasting for Maritime Container Terminals

An evaluation of two approaches based on operation plans

Norman Ihle
R+D Division Energy
OFFIS e.V. – Institute for Information Technology
Oldenburg, Germany

Axel Hahn
Department of Computing Science
Carl von Ossietzky University of Oldenburg
Oldenburg, Germany

***Abstract*— Short-term load-forecasting for individual industrial customers has become an important issue, as interest in demand response and demand side management in modern energy systems has increased. Integrating knowledge of planned operations at industrial sites into the following day's energy-consumption forecasting process provides advantages. In the case of a maritime container terminal, these operation plans are based on the list of ship arrivals and departures. In this paper two different approaches to integrating this knowledge are introduced: (i) case-based reasoning, similar to a lazy-learner that uses available knowledge during the forecasting process, and (ii) an Artificial Neural Network that has to be trained before the actual forecasting process occurs. The outcomes show that integrating more knowledge into the forecasting process enables better results in terms of forecast accuracy**

Keywords— short-term load-forecasting; case-based reasoning; Artificial Neural Networks; maritime container terminals

I. Introduction

Before liberalization of the electricity market, optimizing power demand for industrial enterprises with high-energy consumption was a matter of importance. At that time, grid operators introduced peak load based tariffs and varying day and night rates to encourage peak load avoidance and to shift demand to time-slots that offered lower prices.

In many industrial sectors, electricity procurement has received renewed attention, particularly since the introduction of e-mobility. Maritime container terminals employ a large number of electrified handling equipment and heavy-duty battery powered electric vehicles [1]; therefore, they are currently exploring alternative methods to optimize energy demand and to participate in demand response programs. In addition to an increase in the relevance of energy demand to operational strategies, flexibility in energy demand has also become an emerging issue. Flexibility refers to the possibility of shifting loads. For example, for a container terminal that uses e-mobility, flexibility means having the opportunity to select the most efficient time (range) for recharging a vehicle's battery.

Flexibility requires knowing the container terminal's expected power demand to ensure economic returns. The load curve provides this information: it represents the course of power consumption and consists of 96 values, taken at 15-minute intervals throughout the day.

Simple forecasting methods, based on reference days, can easily be applied. Although just using the daily load curve as a forecast value, regardless of whether it is from the past week or the previous year, limits prognosis accuracy in a highly dynamic environment like a container terminal. Over time, many sophisticated methods for short-term load-forecasting (STLF) have been developed and are now used by utilities. Some of those methods were developed for specific scenarios; others follow a more general approach. Elaborated reviews of different approaches can be found in [2] or [3]. Most were developed and evaluated for forecasting power consumption within distribution grids or for aggregated groups of users. A review of available scientific literature revealed that there has been no application of these load-forecasting methods for container terminals – which have constantly changing, non-periodical handling and transport processes. Although more general methods can be applied to all kinds of energy consumers, there is no systematic evaluation as to which method yields the best results when applied to a single industrial site in the logistics domain. In addition, research about integrating operation plan knowledge into the forecasting process is also lacking. Since the consumption of electricity at a container terminal is dependent upon factors such as the number of containers being handled, a solution based on case-based reasoning (CBR) is proposed. The idea is that days with similar logistic operation requirements have similar load consumption patterns. This is due to the fact that a few specific handling devices have a great influence on the total energy consumption of the terminal. Artificial Neural Networks (ANN) on the other hand are known to be able to approximate non-linear functions and to solve problems where the input-output relationship is neither well defined nor easily computable – because ANNs are data-driven. Three-layered feedforward ANNs are especially suited for forecasting; implementing nonlinearities using sigmoid activation functions for the hidden-layer; and linear functions for the output layer [4]. This paper investigates the two different approaches and outlines how terminal operation plans can be integrated into the forecasting process. The remainder of the paper is organized as

P. Haber et al. (Hrsg.), *Data Science – Analytics and Applications*, http://doi.org/10.1007/978-3-658-19287-7_3

follows. Chapter II introduces the sailing list of a terminal that represents the operation plans. It describes how data can be converted into a daily view of the operations. Chapter III introduces the approach based on case-based reasoning and Chapter IV the approach based on Artificial Neural Networks. After discussing how to integrate data from operation planning, both approaches are evaluated regarding their forecasting results and further aspects in Chapter V before Chapter VI gives an outlook on further work.

For the evaluation of the different approaches, we used the well-known error metric mean absolute percentage error (MAPE) throughout the paper:

$$MAPE = \frac{100}{N} \sum_{t=1}^{N} \frac{|F_t - A_t|}{A_t} \quad (1)$$

A_t is the actual metered value and F_t the forecasted one at time t. N describes the number of values. In our case, since we forecast the load curve for one day, N equals 96 (one value for every 15 minutes).

II. Daily Operations and Power Consumption at a Maritime Container Terminal

The main task of a maritime container terminal is to organize the container handling of container ships. The ship operation or berthing area is equipped with quay cranes for loading and unloading of vessels. Import as well as export containers are stocked in a yard, which is divided into a number of blocks. The container in the block storages are handled by automatic rail mounted cranes that pick up, stack and sort the containers during the storing process.

Reefer containers need an electrical supply for cooling, so some block storage areas are especially equipped for their needs. The truck and train operation area provides the interface to transportation systems outside of the terminal, the so-called hinterland [5].

The terminal's daily operations are planned using the so-called sailing list. This list includes the information of scheduled ship arrivals and departures and additional information on the number of containers that have to be handled with each ship. Figure 1 shows an excerpt of a sailing list.

Since the electricity consumption of the terminal is highly related to the number of container handlings, the sailing list can be used for the integration of operation information into the STLF-process. Since the goal is to forecast the load curve for exactly one day, it is necessary to transform this list into

JSNR	Ship Name	Ship Type	Expected Arrival	Expected Departure	Loading	Un-loading
308505	AKACIA	Feeder	04.09.2013 15:45	05.09.2013 04:00	373	244
306757	OOCL KAOHSIUNG	ATX	05.09.2013 00:10	05.09.2013 15:35	1399	16
308442	A LA MARINE	Feeder	05.09.2013 07:00	06.09.2013 06:00	534	556
308632	KAHN LAUK	Kahn	05.09.2013 15:55	05.09.2013 16:30	0	5
306926	EMOTION	Feeder	06.09.2013 07:45	07.09.2013 00:55	458	333
307896	APL VANDA	LOOP_7	06.09.2013 17:55	08.09.2013 14:00	2524	3031
308543	LEONIE P	Feeder	06.09.2013 15:45	06.09.2013 20:15	22	73

Figure 1: Excerpt of a sailing list operation schedule

information for exactly one day. E.g. the JSNR 307896 has a berthing time with a three-day range. We split up the information into three virtual berthing entries, one on each day of the berthing time:

- Expected Arrival 06.09.2013 17:55,
  Expected Departure 06.09.2013 23:59
- Expected Arrival 07.09.2013 00:00;
  Expected Departure 07.09.2013 23:59
- Expected Arrival 08.09.2013 00:00,
  Expected Departure 08.09.2013 14:00

The information about the container numbers has to be spl up as well. Using the simulation model described in [6] it ca be shown that the container-handling rate for one ship drop over time. This can be approximated using the followin formula where n is the number of full hours the ship is berthin and C describes the number of containers to be handled:

$$C_{h_i} = (1{,}3 - \frac{0{,}6}{h_n - 1} * (h_i - 1)) * \frac{C_{total}}{h_n} \quad with\ h_i = 1, \dots, h_n \quad (2)$$

In addition to the number of containers handled, the day o the week also has a remarkable impact on power consumptio Although the terminal is operating 24/7, records show that day during the week (Monday to Friday) have slightly differe power consumption characteristics when compared to th weekend (Saturday & Sunday). On most weekdays, change o shifts can clearly be recognized three times a day; while o weekends the up- and downturns of the load curve are mo irregular having a slightly lower overall level in most case This might be due to fewer administrative staff working in th office building and German laws (case study terminal locatio prohibiting truck driving on Sunday. On some holidays th terminal is closed; this has an impact on the day before an after the holiday because some shifts may be canceled. Figu 2 shows some examples of load curve characteristics o different types of days[1]. Load curve characteristics we calculated according to day type (work day: Monday throug Friday or weekend: Saturday and Sunday).

Besides the weekday characteristic, data shows that weath influences the load curve. Power consumption is generall higher on days with low temperatures because energy is neede for additional heating purposes, at least in container termina

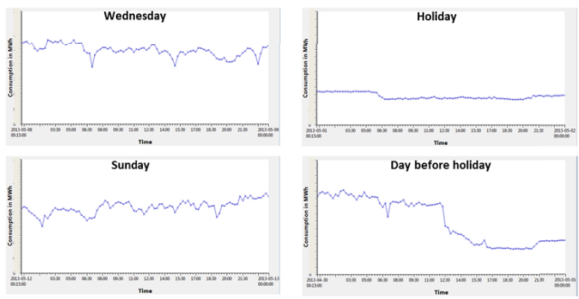


Figure 2: Load characteristics of different day types

[1] Due to reasons of confidentiality the scales of the load curves are not shown in load curve diagrams throughout the paper

located in Northern Europe. Lighting needs also increase on days in winter compared to summer due to late sunrise and early sunset. Lower temperatures in winter reduce reefer container power consumption requirements. However, this effect cannot be generalized to all container terminals, since terminals in regions with higher temperatures might use more power for cooling purposes on hot days in summer.

III. Case-Based Reasoning Approach

Since it can be shown that the consumption of electricity at a container terminal relies on the number of containers being handled, we propose a solution based on case-based reasoning (CBR). The concept of CBR is that similar problems have similar solutions. A case is usually defined by a problem description and a corresponding solution. Aamodt und Plaza introduced a process model, the 4-R cycle (see Figure 3) that has been used as a CBR-reference model up until today [7]. It describes the process of case-based reasoning in 4 steps: retrieve, reuse, revise, retain. The core of the model is the case base that stores verified cases from the past. In a first step the current problem is used to retrieve a case with a similar problem description from the case base. The k-nearest neighbor algorithms are widely used for this similarity-based search. The solution is then adapted to reuse the similar solution for the current problem. This suggested solution is then revised before it is retained in the case base if it was successfully tested.

For the container terminal case, the CBR is aligned with the fact that days with similar logistic operation requirements have similar load consumption patterns. For the purpose of short-term load-forecasting we decided to use a structural approach for the case representation. The idea underlying the structural approach is to represent cases according to a structured

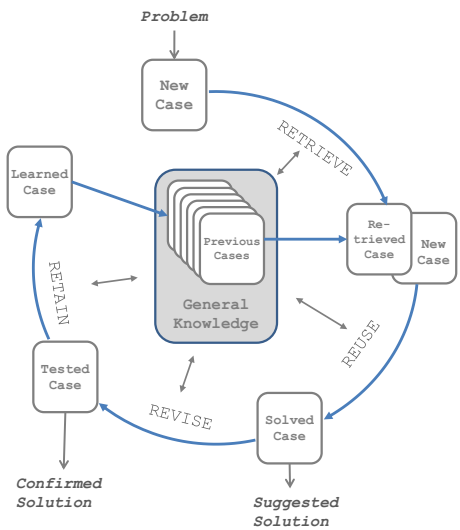


Figure 3: Case-Based Reasoning cycle [7]

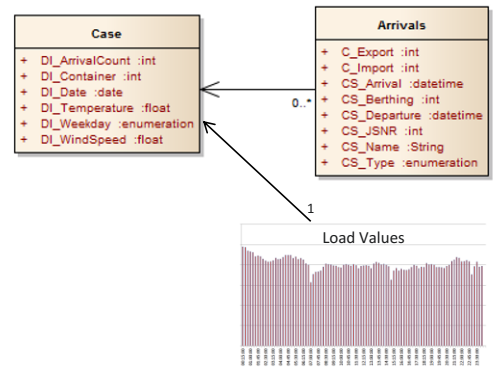


Figure 4: Case format

vocabulary or a domain model, often represented by attribute value vectors [8].

In the CBR approach the case description is composed of two layers. The first layer consists of information on scheduled ship arrivals and departures, and the corresponding container handling numbers for each arriving and departing vessel as described in the sailing list (see Figure 1). The second layer adds information that applies for the whole day like weather data and day of the week information. Figure 4 shows a class diagram that represents the case format. Each case includes information pertaining to the ship's arrival and each case is linked to the corresponding day's load curve. The power consumption values are not stored within the case, but rather in an additional database. Since the values are not needed for similarity assessment, this keeps the case structure straightforward.

Because the berthing time of the ships might exceed date limits, we use the daily-delimited information as described in chapter II. The second layer combines information about ship arrivals and departures with information about temperature, wind-speed, weekday and any additional information that can be derived from the sailing list, such as number of arrivals and the overall number of container handlings. For each of the attributes an individual similarity-measure is assigned to compute the similarity between the same attributes of two cases. For example, when comparing two cases the arrival and departure time of each ship plays an important role in assessing similarity. In the sailing list the information is available in a date-time format, the date being separated from the time by a blank. Since the single cases represent a daily view of the terminal operations, the date information can be disregarded; only the time information is relevant for similarity calculations. In order to be able to use existing similarity measures, the time value is converted into an integer value. This value represents the number of minutes that have passed since the start of the day. The time value 00:00 is represented by 0 and the end of the day at 23:59 by 1439. If the absolute distance between the time values of two different arrival-attributes is less than 60 (minutes) both arrivals were not more than an hour apart on the same day and are therefore quite similar. Similarity decreases quickly as the time difference increases. To represent this, a

sigmoid similarity function based on the distance between the case and the query value is chosen as described by (3).

$$sim_{CS_{Arrival}}(d_{CS_{Arrival}}) = \frac{1}{e^{\frac{d-120}{30}} + 1} \quad (3)$$

The similarity values of the single attributes are then aggregated to an overall similarity value for these two cases using a weighted sum. When aggregating the values of the second layer to compute the overall similarity, the aggregated similarity measure of the first layer is regarded as one value. Each case in the case base points to the metered load curve of the corresponding day as part of the solution.

The adaptation process has three parts that compare: differences in temperature by day; the number of containers handled per hour; and the average yearly increase in units of electrically powered handling equipment (during the past three years). The first part of the adaptation process acknowledges that electricity consumption is generally higher if the temperature is low. Therefore temperature differences between the current day and the most similar case day are compared. The second part of the adaptation process recognizes that if the number of containers handled in an hour, in the current case, is significantly higher than the ones in the most similar case – the electricity consumption of the retrieved load curve is raised during that hour by a factor dependent on the difference. More details about the adaptation process can be found in [9].

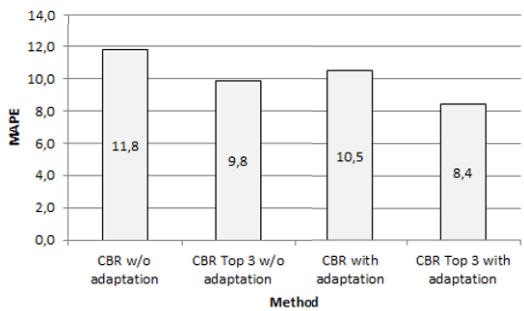


Figure 5: Results of the different CBR approaches

Available historic consumption data show an increase in average power consumption over time. This is a result of the evolving electrification of handling equipment. For example, new battery-powered heavy-duty container transport vehicles have been integrated into terminal operations. Therefore the third part of the adaptation process includes a factor that increases the forecasted values with regard to the average yearly increase (over a three year time period) in the amount of electrically powered handling equipment in use. The result is a new load curve that represents the forecast for the requested day based on the (adapted) load curve of the most similar day. In an additional step, the adapted load curves of the three most similar cases can be arithmetically averaged in order to smooth outliers out of the forecast. This approach will be referred to as the Top 3 approach, while the CBR approach refers only to the adapted result of the most similar case.

Figure 5 shows the average results using these two approaches. It can be seen that the adaptation improves both results.

IV. Artificial Neural Network Approach

Artificial Neural Networks (ANN) are inspired by the structure of the human brain. They consist of a large number of parallel processing units. These neurons are quite simple units that are connected to each other. These connections can be activated using given rules. Each connection from neuron i to neuron j has an individual weight w_{ij} that is adjusted during the training process that is needed to prepare the network for its task. Each network consists of an input-layer, which receives the input data, a number of hidden layers that are responsible for the computation, and an output-layer for the result. Each neuron has an activation function that is responsible for taking the weighted sum of the inputs and calculating the corresponding output for the neuron. Different ANN-structures and training algorithms have been proposed over time. Figure 6 shows a generic structure of a Feedforward Artificial Neural Network with one hidden layer. The values x_1 to x_n describe the input vector and y_1 to y_n the output vector.

Forecasting with ANNs involves two major steps: training and learning. Training of feedforward networks is usually performed in a supervised manner. It is assumed that a training set is available, given by the historical data and containing both inputs and the corresponding desired outputs, which is presented to the network. During the learning process an ANN constructs an input-output mapping. The weights for all neurons, within the hidden and the output layers, are adjusted after every iteration. This adjustment is based on the minimization of an error measure between the output produced and the desired output. The error minimization process is repeated until an acceptable criterion for convergence is reached.

Using an ANN for forecasting the short-term power consumption at a maritime container terminal, the number of container movements per quarter hour can be used as input. Each quarter hour of a day is represented by an input neuron. To calculate the container movements for each hour we use (2) to calculate the hourly values of container handlings. For each

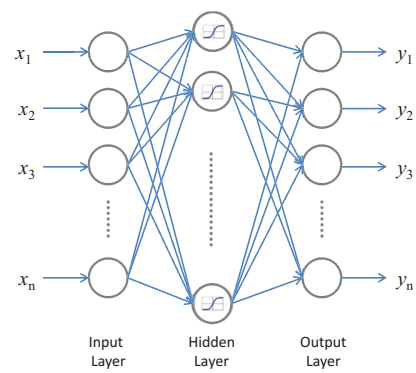


Figure 6: Generic structure of an Artificial Neural Network

hour this number is then divided by 4 to determine a daily total of 96-quarter hour values; this is the same resolution as the load curve we want to forecast. It is necessary to use the same value for each quarter of an hour, since more detailed information about the number of containers handled is not available. In a first step container handling numbers of each quarter hour of the day were used as input. The corresponding power consumption values of each quarter hour represent the intended output. We used a sigmoid function as an activation

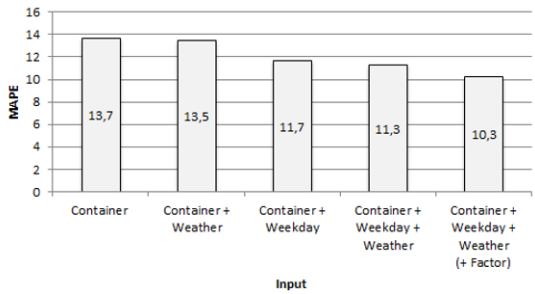


Figure 7: Results of the different ANN approaches

function, since we had normalized the values to the range between 0 and 1 and had no negative values. For training purposes, we used a resilient back propagation algorithm with a learning rate of 0.5. The inputs were the daily number of containers and the corresponding power consumption values of the years 2010 to 2012. The forecasting results were tested with values from the year 2013, which were not part of the training data set. Starting with a structure of 96 input neurons, 192 neurons in the hidden layer and 96 output neurons, we successively added new input neurons that represented information about the day of the week and weather values as temperature and wind-speed for the day. For each new input neuron we added two neurons to the hidden layer. Every time we compared the results to the results of the previous networks. As expected, the average error rates dropped when more information was presented to the network (see Figure 7). The input of container handling numbers, weekday information, and weather yielded the best results. In addition, the results improved when an adaptation factor was used during training. This factor represented the annual average increase in power consumption. We applied the factor to the desired training set

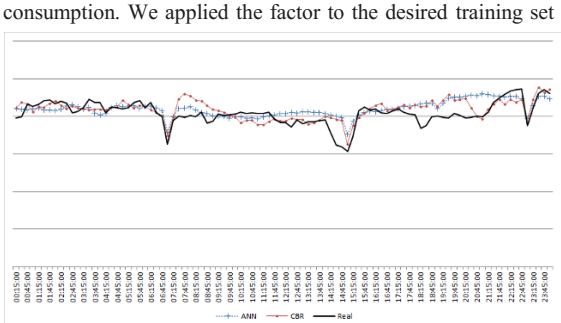


Figure 8: Forecast and real values for May 24th

results, so that they represented a more up-to-date level of power consumption at the container terminal. Adding more neurons in the hidden layer or even adding a new hidden layer did not improve the forecasting quality of the Artificial Neural Network.

V. Evaluation

For the evaluation of both approaches, we used the year 2013 as reference year. While the ANN was trained with data from the year 2010 to 2012, the case base of the CBR approach consisted of cases from each day in 2010 to 2012 and each case in 2013 until two days before the forecasting date, so it grew within the year. We compared the forecasted values for each day to those metered during the respective day. Both approaches showed similar results on the first iterations during the development. While the ANN with just the quarter hour container handling numbers as input yielded an average MAPE of 13.7 for the year 2013, the CBR approach without adaptation yielded 11.8. The forecasting results of the ANN could be improved by adding additional information about the weekday (average MAPE for 2013: 11.7) or weather data (average MAPE for 2013: 13.5). Adding both the weekday and the weather characteristics the forecasting results offered the best result. When applying an adaptation factor during training, the results could be improved even further. Using the CBR approach, the initial result could be improved by introducing adaptation. The result was improved remarkably by using the average of the three most similar adapted load curves as the forecasted load curve. Here the MAPE for 2013 was as low as 8.4 percent. Only in October 2013 was it slightly higher than 10 percent. Figures 6 and 7 show the MAPE values of both the ANN and the CBR approach when using different parameters as input values or using adaptation within the CBR approach.

Figure 8 shows the forecasting result for May 24th 2013. Both the ANN and the CBR approach yield a MAPE of about 5, the CBR Top 3 with 5.1 slightly better than the ANN with 5.6. In this example the forecasting results approximate the real values in the early hours of the day quite well. While the CBR forecast still includes a lot of small peaks and valleys over time, as do the real values, the ANN forecast describes a smoother progress in the power demand. After the first change of shifts at 07:00h the CBR approach forecasts a rise in power consumption that did not occur, but in the following hours the forecast values converged with the real values. After the second change of shift at 15:00h (which seemed to have started a little bit earlier than usual considering the real values) both the ANN and the CBR approach forecasted an increase in power consumption that did not occur at that time, but later (at around 21:30h). This might be due to a ship arriving later than scheduled or just a delay in the start of the container handling for a ship.

Figures 9 and 10 show the monthly MAPE values of the ANN and the CBR approach together with the range of daily MAPE values over the month. In comparison it was evident that not only the average MAPE values are lower when using the CBR Top 3 approach, but also value fluctuation is also lower in the CBR case. While the MAPE values of the CBR approach never exceed a value of 25, the ANN approach did during eight months in 2013.

The results also showed that it is difficult to forecast the load curves on holidays and those days before and after the holidays. On the one hand there are too few cases or examples of these days in the past data (each year has only 6 holidays), on the other hand these examples differ in their characteristics so that it is hard to learn patterns from them. For the holidays themselves, the results were acceptable, since only a base consumption without a high fluctuation was recorded. But with regards to the days before and after the holidays, the times of stopping or resuming operations varied noticeably. Hence,

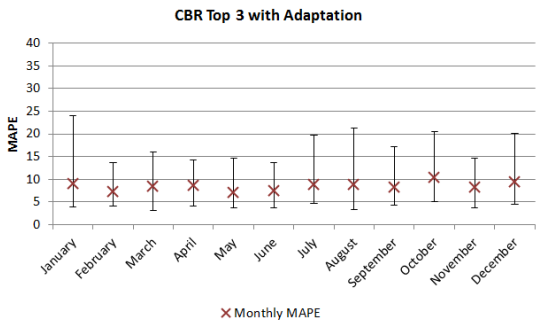


Figure 9: Monthly results of the CBR approach

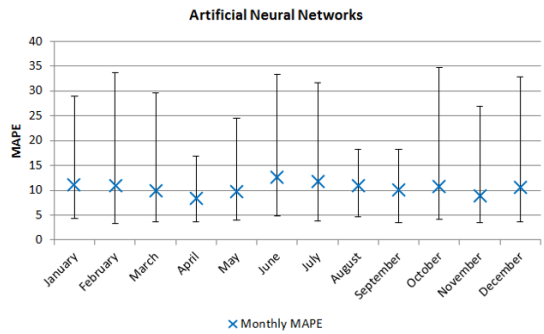


Figure 10: Monthly results of the ANN approach

using the power consumption values of the same holidays from previous years yielded more accurate results than the forecasts based on the CBR and the ANN approach.

VI. Conclusion & Further Work

We have shown that case-based reasoning, as well as Artificial Neural Networks can be used for knowledge-based short-term load-forecasting at a maritime container terminal. The knowledge is based on the schedule of ship arrivals and departures and the corresponding expected number of container handlings. It must be noted that besides the hourly split-up of the number of container handlings, the Artificial Neural Network approach needed no further modeling of any knowledge structure or other container terminal processes. In contrast, the CBR approach required a more knowledge intensive modeling phase. Besides the case structure that is in parts aligned to the structure of the sailing list, knowledge about terminal operations is also contained in the similarity measures as well as in the adaptation rules. This also offers an advantage, since the modelled knowledge can be used to explain to the user how the result was computed. The dates of the most similar days used for the forecasting can be presented as well as the applied rules. The Artificial Neural Network on the other hand represents a black-box, which hardly needs to be adapted when applied to other container terminals or even to other logistic systems.

Case-based reasoning is a promising approach to integrating knowledge about container terminal operations into the short-term load-forecasting process. Further work has to be invested into building up the case base. Cases of days with some hours without terminal operation (e.g. due to maintenance activities) should be distinguished from the case base. Knowledge about upcoming maintenance activities could be integrated into the adaptation process. Regarding the ANN approach, different types of network structures, such as recurrent neural network, still need to be investigated as to any advantages that they might provide for the forecasting or the training process. Until now, only feed-forward neural networks have been applied.

References

[1] J. Schmidt, L.-P. Lauven, N. Ihle and L. M. Kolbe, “Demand side integration for electric transport vehicles,“ *International Journal of Energy Sector Management,* pp. 471 - 495, 9 2015.

[2] T. Hong, “Short Term Electric Load Forecasting,“ North Carolina State University, Raleigh, 2010.

[3] A. K. Singh, I. S. Khatoon, M. Muazzam und D. K. Chaturvedi, „An overview of electricity demand forecasting techniques,“ *Network and Complex Systems,* pp. 38-48, 3 2013.

[4] J. Catal□o, S. Mariano, V. Mendes und L. Ferreira, “An artificial neural network approach for short-term electricity prices forecasting,“ in *International Conference on Intelligent Systems Applications to Power Systems*, IEEE, 2007.

[5] H.-O. G□ntherand K.-H. Kim, “Container terminals and terminal operations,“ *OR Spectrum,* pp. 437-445, 2006.

[6] N. Grundmeier, N. Ihle und A. Hahn, “A Discrete Event-driven Simulation Model to Analyse Power Consumption Processes in Container Terminals,“ *Simulation in Production and Logistics, Fraunhofer IRB Verlag, Stuttgart,* 2015.

[7] A. Aamodt und E. Plaza, “Case-Based Reasoning: Foundational issues, methodological variations, and system approaches,“ *AI communications* , pp. 39-59, 1994.

[8] R. Bergmann, “Experience management: foundations, development methodology, and internet-based applications,” Springer-Verlag, 2002.

[9] N. Ihle, “Case Representation and Adaptation for Short-Term Load Forecasting at a Container Terminal,” *24th International Conference on Case-Based Reasoning (ICCBR) 2016 - Workshop Proceedings*; 2016

Data Analytics in Community Networks

Beyond Spectral Clustering: A Comparative Study of Community Detection for Document Clustering

Christian Backhage, Kostadin Cvejoski
César Ojeda and Rafet Sifa,
Fraunhofer IAIS
St. Augustin, Germany

Christian Backhage and Rafet Sifa
University of Bonn
Bonn, Germany

***Abstract*—Document clustering is an ubiquitous problem in data mining as text data is one of the most common forms of communication. The richness of the data require methods tailored to different tasks, depending on the characteristics of the information to be mined. In recent years graph-based methods have appeared that allow for hierarchical, fuzzy, and non Gaussian density features to identify structures in complicated data sets. In this paper we present a novel methodology for the clustering of documents based on a graph defined over a vector space model. We make use of an overlap hierarchical algorithm and show the equivalence of our quality function to that of Ncut. We compare our method to spectral clustering and other graph-based models and find that our method provides a good and flexible alternative for news clustering, whenever fine grained details between topics are required.**

I. Introduction

Text mining and text modeling remain important topics of research as the enormous amount of available text data requires methods and techniques to uncover relevant hidden information. In order to find patterns in text, one must initially define a proper model of text and the results obtained will be severely constrained by the flexibility and descriptive power of the model at hand. Several, rather different ideas have proven to work well in this regard. Classical information retrieval (IR) methods define vector spaces over documents with weights depending on word frequency and information metrics. The term frequency inverse document frequency model **tf-idf** is one such approach [8]. Neural network models, such as **doc2vec** [6], define vectors over words and documents from the weights of a trained feed forward or recurrent neural network model which aims at predicting the successive words in a document, provided that the input data is a set of surrounding words. On the other hand, generative probabilistic models such as LDA [2] have been developed in which documents are assumed to result from distributions which describe the statistical behavior of words in text collections. Although probabilistic and neural network methods have been proven to outperform information retrieval methods, we can still improve the descriptive power of the IR perspective through a better study of their vector space structure. In this work, we propose a methodology for document clustering where a graph is defined over the vector space model and use community detection algorithms in this graph, which deliver clustering results similar to spectral clustering methods [11]. This approach has all the know advantages of spectral clustering methods e.g. the detection of clusters which posses a non gaussian density and complicated topology. Furthermore, community detection algorithms allow for the detection of clusters with different densities in the same detection. We make use of a community detection algorithm which also achieves hierarchical clustering and offers a solution for the model selection problem of the number of clusters. This methodology also has the advantage of improving the interpretability of the documents [1] in the vector space model, as connections to different structures are analogous to different topics in probabilistic generative models. In order to prove the feasibility of the methodology for spatial clustering, we provide a comparative study between community detection methods and traditional laplacian algorithms.

The remainder of this paper is organized as follows. We summarize the steps of natural language processing, the methodology for the definition of the network of documents and the community detection algorithm in section II. In section III we discuss the relationship between our algorithm and the Ncut approach of spectral clustering [11] and explain the behavior of the results as related to generative probabilistic models approach such as LDA. Then, as a use case, we show in section IV the results of our methodology for the clustering of a set of posts from a news blogs. Finally, conclusions are drawn in section V.

II. Methodology

The aim of any clustering algorithm is to group a given set of data points into clusters of *similar characteristics*. In order to achieve this on a data set which contains N different points $\mathcal{D} = \{x_1, x_2, ..., x_N\}$, we define a similarity function over $\mathcal{D}$ which allows us to define a function to quantify the cluster quality. This function gives formal meaning to the principle of *similar characteristics* and clustering can then be understood as the task of maximizing this objective function.

Spectral clustering works by defining where each node represents a data point. The objective function is defined on the graph, following the heuristic that groups of nodes that are more connected to themselves than to the rest of the graph should be grouped together. Although not directly specified, the optimal objective is known to be attained trough the analysis of the eigenvectors of the Laplacian of the graph. This Laplacian is a matrix derived from the graph adjancency matrix and the different degrees of the nodes. In the present work, we propose to tackle the clustering of the nodes with the use of methods from community detection in graphs. To this aim, we provide a comparative study between spectral clustering

P. Haber et al. (Hrsg.), *Data Science – Analytics and Applications*, http://doi.org/10.1007/978-3-658-19287-7_4

and two widely known community detection algorithms: The Newman-Girvan [3] algorithm which is based on a null model approach, and the Overlap Hierarchical algorithm [5], a greedy procedure which allows for soft and hierarchical clustering. We provide the equivalence between the objective functions of the Overlap algorithm and that of Spectral clustering. We also improve the efficiency of the Overlap algorithm and apply it to document clustering as a use case example. We begin with introducing basic terminology and then define the objective function of the different algorithms.

A. Concepts from Graph Theory

Consider a network or a graph $\mathcal{G} \equiv (\mathcal{E}, \mathcal{V})$, where $\mathcal{V} \equiv \{v_1, ..., v_n\}$ is a set of vertices (or nodes) and $\mathcal{E} \subseteq (\mathcal{V} \times \mathcal{V})$ is a set of edges (or links). $|\mathcal{V}| = n$ defines the total number of nodes. The edge structure of the graph can be encoded in the adjacency matrix $[W] = w_{ij}$. For undirected sparse graphs the values of w_{ij} is one if there exists an edge between node i and j and zero otherwise. Dense and asymmetric graphs can be defined depending on the similarity measure employed. The degree of a node is defined as $k_i = \sum_j w_{ij}$. For sparse undirected graph, this represents the number of edges incident to node i.

A subgraph or $\mathcal{C} \equiv (\mathcal{E}(\mathcal{C}), \mathcal{V}(\mathcal{C}))$ of a graph $\mathcal{G}$ is defined by a set of nodes $\mathcal{V}(\mathcal{C}) \subseteq \mathcal{V}$ and the edges $\mathcal{E}(\mathcal{C}) = \{(v_i, v_j) : v_i, v_j \in \mathcal{V}(\mathcal{C}) \wedge (v_i, v_j) \in \mathcal{E}\}$. The number of nodes of the subgraph is given by $|\mathcal{C}| = n_c$. The group of nodes of a subgraph define a community or cluster within a graph and a partition of a graph is defined as a set of subgraphs such that:

$$\mathcal{P} \equiv \{\mathcal{C}_1, ..., \mathcal{C}_M\} \text{ such that } \bigcup_{i=1}^{M} \mathcal{V}(\mathcal{C}_i) = \mathcal{V}(\mathcal{G}) \tag{1}$$

In the case of hard clustering the subgraphs are disjoint in the sense that $\mathcal{V}(\mathcal{C}_i) \cap \mathcal{V}(\mathcal{C}_j) = \emptyset \; \forall \mathcal{C}_i, \mathcal{C}_j$. Soft or fuzzy clustering on the other hand allows for communities to share nodes. In this case, the partition is known as a *cover* and the the objective function for clustering is simply a function such that $\phi : \mathcal{P}, \mathcal{C}_\rangle \rightarrow \Re$ defining the quality of the partition.

B. Spectral Clustering

Spectral clustering defines a family of algorithms which achieve graph clustering based on graph Laplacian matrices. The objective is to partition the nodes by means of clustering in the space defined by the eigenvectors of the chosen Laplacian. In the following, we make use of the *unnormalized graph laplacian*, defined as

$$L = K - W \tag{2}$$

where K is the diagonal matrix defined from the degrees of the nodes such that $K_{ii} = k_i$ and zero elsewhere. To partition the data into M groups, we cluster the N data points for the nodes defined via the N entries of the M eigenvectors of L via the k-means clustering algorithm. The objective function minimized trough this procedure is know as the *NCut*, defined below in section III.

C. Newman- Girvan Algorithm

The Newman-Girvan algorithm is a global algorithm in that its objective function depends on the graph as a whole and is not a local measure of the nodes of a given subgraph. It follows a null model heuristic where, given a particular community in a partition, it considers the difference between the current number of edges within the community and number of edges such partition could have achieved randomly. The larger this difference, the better the community. The Newman Girvan objective function is [1] and encodes this difference as

$$Q = \frac{1}{2m} \sum_{ij} \Big(w_{ij} - \frac{k_i k_j}{2m} \Big) \delta(\Gamma_i, \Gamma_j) \tag{3}$$

where Γ_i indicate the community to which node i belongs and $\delta(A, B) = 1 \iff A = B$ 0 otherwise.

D. Overlap Hierarchical Algorithm

The *internal degree* of a node $v \in \mathcal{C}$ ($k_v^{int}(\mathcal{C})$) is defined as the number of edges with nodes in $\mathcal{C}$. On the other hand $k_v^{ext}(\mathcal{C})$ is the number of edges of a node $v \in \mathcal{C}$ with nodes not in $\mathcal{C}$. We also define for the subgraph $\mathcal{C}$, the internal(external) degree

$$k_{\mathcal{C}}^{int(ext)} = \sum_{v \in \mathcal{C}} k_v^{int(ext)}(\mathcal{C}) \tag{4}$$

and observe that $k_{\mathcal{C}}^{int(ext)}$ equals twice the number of edges for nodes in $\mathcal{C}$.

A community then is a subgraph $\mathcal{C}$ such that the values of $k_{\mathcal{C}}^{int}$ arelarge whereas $k_{\mathcal{C}}^{ext}$ is small. In order to quantify this heuristic, we introduce the *fitness* $f_{\mathcal{C}}$ of a subgraph $\mathcal{C}$ which forms the basis for the *overlapping hierarchical method OH* [4]. In particular, we work with

$$f_{\mathcal{C}} = \frac{k_{\mathcal{C}}^{int}}{(k_{\mathcal{C}}^{int} + k_{\mathcal{C}}^{ext})^{\alpha}} \tag{5}$$

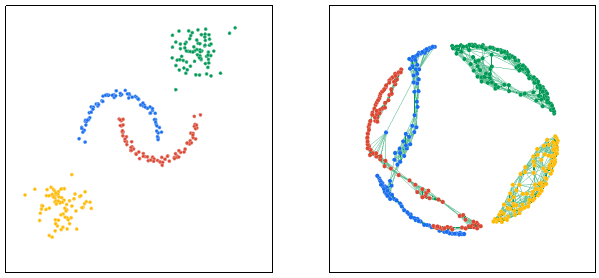
With this fitness function, we are searching for some form of cohesion in the subgraph $\mathcal{C}$, as the less edges there are from $\mathcal{C}$ to the outside of the subgraph, the higher the fitness. Big fitness means a good community. The **OH** algorithm will try to search for groups of nodes with high fitness value. The parameter α is introduced as way of tuning the *scale* at which we are searching for the community. This parameter will allow us to study the different hierarchies if existent. The larger α, the more the connections to the outside of the subgraph ($k_{\mathcal{C}}^{ext}$) will impact the fitness. Consequently, we cannot improve the fitness by adding many nodes.

We also define the node fitness $f_{\mathcal{C}}^v$ with respect to a subgraph $\mathcal{C}$

$$f_{\mathcal{C}}^v = f_{\mathcal{C}+\{v\}}^v - f_{\mathcal{C}-\{v\}}^v \tag{6}$$

This fitness is just a variation of the subgraph fitness $f_{\mathcal{C}}$ between the graph with and without node v. When implemented, this algorithm will find *covers* of the graph.

[1] also known as modularity



(a) spatial data (b) associated k-NN graph

Fig. 1. Data sampled from two moon-shaped clusters and two Gaussian clusters used to compare the performance of different clustering algorithms.

The algorithm contains two main steps, initially we obtain the natural community for one given node and use this procedure iteratively to find the whole cover. We refer to this initial node for the community detection as the *seed* node v_s. We start by defining the subgraph $\mathcal{C}$ with only this node i.e. $k_{\mathcal{C}}^{int} = 0$. We then define the set of neighbors of community $\mathcal{C}$ as $\Gamma(\mathcal{C}) = \{u | u \in \mathcal{V}$ and $u \notin \mathcal{C}$ and $(u, v) \in \mathcal{E}\}$ i.e. the neighbors of nodes in $\mathcal{C}$ not in $\mathcal{C}$. The algorithm works by including into the community $\mathcal{C}$ those neighbors with largest fitness and removing from the community those which have a negative fitness once the new node is included. These removed nodes will define independent communities of size one. We introduce the algorithm in Alg. 1. The algorithm stops when all the neighbors of the community being defined have negative fitness. In order to improve the performance of the algorithm for the selection of the natural communities we use a dynamical programming approach. Notice that in Alg. 1 at steps 4 and 6 there is a modification of the community being detected by the inclusion or the removal of a node. It is desirable then, to store the values of $k_{\mathcal{C}}^{ext}$ and $k_{\mathcal{C}}^{int}$ in order to modify the fitness by only changing these quantities and not accessing the adjacency matrix of the whole group once again. We define a *guest* node of $\mathcal{C}$ as $g \in \Gamma(\mathcal{C})$ which is being tested for inclusion at step 4. After inclusion, the new external degree will be

$$k_{\mathcal{C}}^{ext} \leftarrow k_{\mathcal{C}}^{ext} - k_g^{int} + k_g^{ext} \quad (7)$$

$$k_{\mathcal{C}}^{int} \leftarrow 2((k_{\mathcal{C}}^{int}/2) + k_g^{int}) \quad (8)$$

Also for a test node $t \in \mathcal{C}$, which is considered for removal at step 6, we have

$$k_{\mathcal{C}}^{ext} \leftarrow k_{\mathcal{C}}^{ext} + k_t^{int} - k_t^{ext} \quad (9)$$

$$k_{\mathcal{C}}^{int} \leftarrow 2((k_{\mathcal{C}}^{int}/2) - k_t^{int}) \quad (10)$$

This allow us to obtain the variation of the node fitness as required by Eq. 6, only accessing the neighbors of the guest nodes as given by the adjacency matrix. The degrees of the nodes in the community are kept in memory.

Algorithm 1 detects one community for each seed node v_s. To obtain the full cover, we repeat the procedure selecting seed nodes at random which have not been selected as part of any community yet. For these we let Alg. 1 unfold detecting nodes independently of past detections, allowing a node to belong to different communities at a time. One can thus think of the procedure as a greedy optimization of the fitness function [4].

Algorithm 1 Natural community of a node v_s

Require: Graph $\mathcal{G} \equiv (\mathcal{E}, \mathcal{V})$, v (seed node)

1: $\mathcal{C} \leftarrow \{v\}$
2: **while** there exists a node $v \in \Gamma(\mathcal{C})$ such that $f_{\mathcal{C}}^{v} > 0$ **do**
3: $\quad u = \underset{\hat{u} \in \Gamma(\mathcal{C})}{\operatorname{argmax}} f(\hat{u})$
4: $\quad \mathcal{C} \leftarrow \mathcal{C} \cup \{u\}$
5: $\quad$ **while** there exist $\tilde{u} \in \mathcal{C}$ such that $f_{\mathcal{C}}^{\tilde{u}} < 0$ **do**
6: $\quad\quad \mathcal{C} \leftarrow \mathcal{C} \setminus \{\tilde{u}\}$
7: $\quad$ **end while**
8: **end while**
9: **return** $\mathcal{C}$

E. Histogram Method for Model Selection

We must still develop methods to decide among *covers* of different α. To this end, we follow a simple line of reasoning: the better the cover, the more stable it is in a range of values of α. This implies that, provided the cover is stable i.e. relevant, we should be able to recover the same cover from a wide enough range of α values. In order to characterize each cover (or fuzzy partition) we introduce the *average fitness* of a cover

$$\bar{f}_{\mathcal{F}}(\alpha = 1) = \frac{1}{n_c} \sum_{i=1}^{n_c} f_{\mathcal{C}_i} \quad (11)$$

This fitness is obtained for a fixed value of α regardless of the α used for computing the partition. In order to identify the most stable covers, we perform the community detection for a range of α values. We obtain the histogram of $\bar{f}_{\mathcal{F}}$ values and identify the most stable covers by searching for peaks in the histograms, i.e. we pick the value of the fitness which appears the most and select the value of α which delivered them.

III. Equivalence to Ncut

Similar to our methodology, spectral clustering works by defining a graph over given data points. Clustering is achieved

by minimizing a objective function that considers the eigenvectors of the graph Laplacian [11] [10]. In this section, we prove that our above approach is equivalent to the definition of the Ncut objective function of spectral clustering [10]. Our discussion will make use of the notation in [11]. We define $\text{vol}(\mathcal{C})$ as the total number of edges incident to nodes in $\mathcal{C}$. We use $\text{vol}(\mathcal{C}) = \sum_{i \in \mathcal{C}, j} A_{i,j}$ where the sum over $i \in \mathcal{C}$ implies that $v_i \in \mathcal{C}$. The number of outer edges is defined as $k_{\mathcal{C}}^{out} = W(\mathcal{C}, \bar{\mathcal{C}})$ where $\bar{\mathcal{C}}$ represents all the nodes not in $\mathcal{C}$, $\bar{\mathcal{C}} \equiv \mathcal{V} \backslash \mathcal{C}$ (its complement). The indegree of a community will be defined by $k_{\mathcal{C}}^{int} = \text{vol}(\mathcal{C}) - W(\mathcal{C}, \bar{\mathcal{C}})$. Using this notation the fitness function amounts to

$$f(\mathcal{C}_i) = \frac{\text{vol}(\mathcal{C}_i) - W(\mathcal{C}_i, \bar{\mathcal{C}}_i)}{\text{vol}(\mathcal{C}_i)^{\alpha}} = \frac{1}{\text{vol}(\mathcal{C}_i)^{\alpha-1}} \Big[1 - \frac{W(\mathcal{C}_i, \bar{\mathcal{C}}_i)}{\text{vol}(\mathcal{C}_i)} \Big] \quad (12)$$

For $\alpha = 1$ and a fixed number of communities M, we can define the objective function $F_{\mathcal{P}}$ in order to measure the quality of a graph partition $\mathcal{P} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_M\}$ [2].

$$F_{\mathcal{P}} = \sum_i f(\mathcal{C}_i) = \sum_i \Big[1 - \frac{W(\mathcal{C}_i, \bar{\mathcal{C}}_i)}{\text{vol}(\mathcal{C}_i)} \Big] \quad (13)$$

Notice that the maximum obeys $\max\{F_{\mathcal{P}}\} = \min\{-F_{\mathcal{P}}\}$ for the node assignments in $\mathcal{P}$. For a fixed number of communities M we have

$$\min\{-F_{\mathcal{P}}\} = \min[2\text{cut}(\mathcal{P}) - M] = \min\{\text{cut}(\mathcal{P})\} \quad (14)$$

where we define

$$\text{cut}(\mathcal{P}) = \sum_i \frac{W(\mathcal{C}_i, \bar{\mathcal{C}}_i)}{\text{vol}(\mathcal{C}_i)} \quad (15)$$

By searching to minimize the value of the fitness $f(\mathcal{C}_i)$ in Eq. 12 we are searching for a cover which resembles the partition as obtained from Ncut. This relates the spectral clustering approach to our overlap hierarchical algorithm. The objective function to by optimized is the same for both algorithms provided that the fitness parameter is $\alpha = 1$.

IV. Results

A. Synthetic Data

In order to study the behavior of the different algorithms, we first compare the detection results for a data set whose underlying distribution on $\Re^2$ corresponds to two moon shaped topologies as well as two gaussians. We present the sample and the associated k nearest neighbors graph (kNN) in Fig. 1. This is the graph used in the analysis and it is defined by connecting each data point (node in the graph) with its k nearest neighbors according to the euclidean distance.

We study the quality of the clustering achieved by the different algorithms by modifying the dispersion of the points for both types of clusters and performing the detection with the different algorithms. The goal is to explore how the clustering quality achieved is affected by changes in the dispersion of the data sets. In order to quantify the quality of each detection, we use information theoretic metrics of qualities of partitions. For a data set $\mathcal{D} = \{x_1, x_2, ..., x_N\}$ we introduce the entropy $H(\mathcal{A})$ [7] for a partition $\mathcal{A}$ over $\mathcal{D}$ defined as

$$H(\mathcal{A}) = -\sum_{i=1}^{|\mathcal{A}|} \frac{n_k}{N} \log \frac{n_k}{N} \quad (16)$$

, where n_k is the number of nodes in community k and $|\mathcal{A}|$ is the number of communities or clusters of partition $\mathcal{A}$. This entropy is a measure of the *information* of the data set provided by the partition. [3]. To measure how similar two partitions are, we define the mutual information [7]. We want to know how much common information

$$I(\mathcal{A}, \mathcal{B}) = -\sum_{i=1}^{|\mathcal{A}|} \sum_{j=1}^{|\mathcal{B}|} \frac{n_{ij}}{N} \log \frac{n_{ij} N}{n_i n_J} \quad (17)$$

is shared by the partitions. We introduce a confusion matrix n_{ij}. This matrix defines how many nodes of community i in $\mathcal{A}$ are in community j in $\mathcal{B}$.

We further define the normalized mutual information

$$I_N(A, B) = \frac{2I(A, B)}{H(A) + H(B)} \quad (18)$$

which gives a value of 1 when the partitions are identical and a value of 0 when the partitions are more different. And we introduce the variation of information

$$V(A, B) = H(A) + H(B) - 2I(A, B) \quad (19)$$

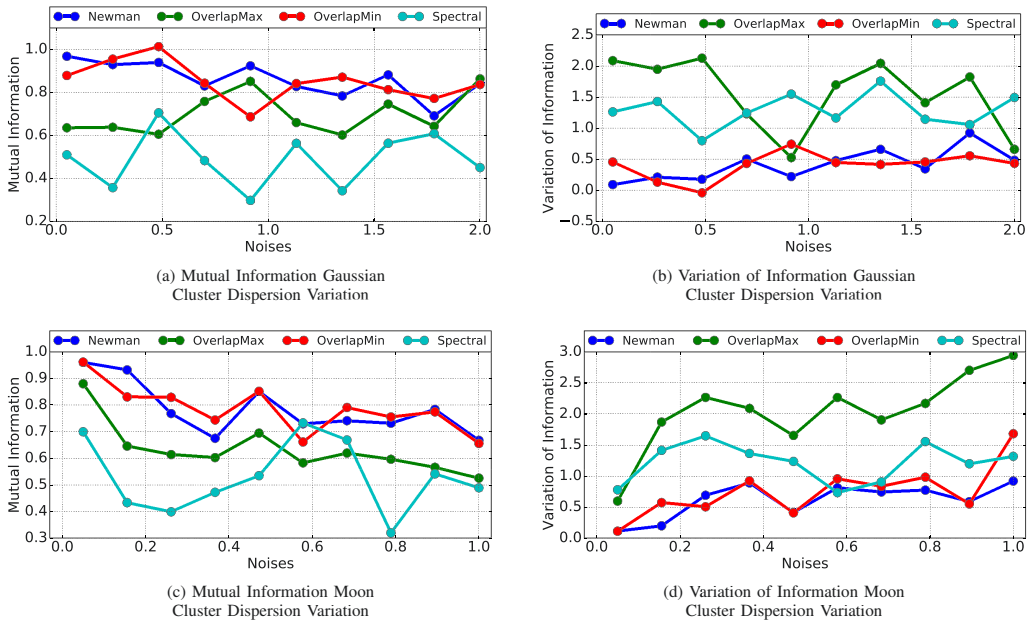
which yields 0 for the exact same partition and larger values for different partitions.

We now perform the detection for a series of samples and calculate the normalized mutual information and the variation of information with the partition obtained by the detection and the partition given by the ground truth of the synthetic samples. We show the results in Fig. 2. For figures 2a and 2b, we modify the dispersion of one of the Gaussian clusters. For figures 2c and 2d we modify the variation of one of the moon clusters. For the detection with the overlap algorithm, the histograms yielded two main fitness factors α. We present the detection with smallest value of α as *OverlapMin* (bigger communities) whereas for the detection with the bigger value of α we write *OverlapMax* (smaller communities).

In both experiments, the community detection approach outperforms spectral clustering with and average mutual information above 0.8. for the Newman and Overlap Algorithm in the variation of the dispersion factor for the Gaussian and over 0.7 for the variation of the dispersion factor for the moon sample. We can interpret this results trough the equivalence between the overlap algorithm and the *MinCut* measure. Both preferred measures of the α values where found for values below 1. This indicates, that, in order to achieve an

[2] In Ncut we require in addition to Eq.1 $\forall(i, j)$ that $\mathcal{V}(\mathcal{C}_i) \cap \mathcal{V}(\mathcal{C}_j) = \emptyset$, hence the name partition

[3] in analogy with the Shannon entropy, when we observe that the probability of a node to pertain to the cluster k is given by $\frac{n_k}{N}$



(a) Mutual Information Gaussian Cluster Dispersion Variation

(b) Variation of Information Gaussian Cluster Dispersion Variation

(c) Mutual Information Moon Cluster Dispersion Variation

(d) Variation of Information Moon Cluster Dispersion Variation

Fig. 2. Quality of the clustering under dispersion variation of different clusters of the synthetic data set.

stable partition, bigger communities are preferred. This implies that the spectral clustering algorithm divided the ground truth clusters into smaller subclusters. Due two the difference in topology, the amount of in between graph edges i.e. $W(\mathcal{C}_i, \bar{\mathcal{C}}_i)$ differs from cluster to cluster. Hence by penalizing the weight of the cluster size trough $\text{vol}(\mathcal{C}_i)^{-\alpha}$ through a small α, we allow for the inter edges to better describe the cluster. This same principle applies to the Newman-Girvan algorithm as its objective function purely considers inter-edges values.

B. Document Clustering

As a practicsal use case, we use our community base clustering framework on a text corpus which comprises 3000 news articles from a political blog (http://www.eclectablog.com/). We extracted the feed data from articles in the period from 2008 to 2014. For the data representation of the documents, we opted for the **tfidf** representation [8] with dimensionality reduction trough Random Principal Component Analysis **RPCA** [9]. This representation was chosen after extensive experimentation with the representation obtained from neural network analysis trough the **doc2vec** model. The neural network model failed to effectively produce a descriptive document space due to the lack of sufficient data to train the model as well as limitations in the text data. Each feed entry only provides a short description of the different news articles, this translated in low poor sample size for the predictive model required by the **doc2vec** approach. Once the **tfidf** and the dimensionality reduction was performed, we generated a graph network trough the k-NN graph methodology for 15 neighbors using the cosine similarity as a distance measure [8]. The histogram was obtained for 500 different detections of the overlap algorithm yielding two main resolutions for $\alpha = 1.01$ and $\alpha = 1.4$. The resolution of bigger communities yielded 507 different communities (including stand alone nodes) as well as 107 communities with more than 10 articles. This shows the wide range of news subjects covered by the site. We show the titles for the two largest resulting communities in Tab. I. The first community refers to news about Barack Obama and the smaller resolution $\alpha = 1.4$ decomposed this cluster into social events involving the US president and news regarding his relationship with the republican party. The other cluster shows information about Obamacare and it was subdivided in relation to republican party criticism and criticism of the law related to insurance policies.

V. Conclusions

In this paper, we proposed a simple methodology for document clustering based on community detection in graphs. The community detection algorithm we utilized provides a solution for the problem of selecting the optimal number of clusters and also allows for detecting fuzzy partitions. Such an approach allows for better interpretability of the results in text clustering since each document can be understood as being related to different topic clusters. Also, different subclusters obtained by our hierarchical procedure provide results that are analogous to those of topic modeling approaches. Each cluster of documents

TABLE I
TWO AUTOMATICALLY DISCOVERED COMMUNITIES FROM OUR ANALYSIS CONTAINING ARTICLES ABOUT THE CURRENT PRESIDENT OF U.S., BARACK OBAMA, AND HIS HEALTH-CARE ACT "THE PATIENT PROTECTION AND AFFORDABLE CARE ACT" A.K.A "OBAMACARE" AND TWO OF THEIR SUB-COMMUNITIES. THE SUB-COMMUNIES CAPTURE THE HIERARCHICAL REPRESENTATIONS IN THE SEMANTIC SPACE, RESPECTIVELY CONTAINING ARTICLES READING OBAMA'S RELATIONS WITH REPUBLICANS AND HIS SOCIAL EVENTS WHEREAS FOR OBAMACARE WE OBSERVE TWO SUBCATEGORIES COVERING ARTICLES ABOUT CRITICS OF THE REPUBLICANS AND INSURANCES.

Com. 1: Barack Obama	
"The Obama administration rules out raising medicaid eligibility age cat food stocks plummet", "This racist snowbilly was almost our vice president palin tweets that Obama is using shuck jive", "Photos and quotes from president Barack Obamas speech in cleveland ohio", "President Obama weighs in on Michigan's right to work for less attack on unions"	
Subcom. 1.1: Republicans vs Obama	**Subcom. 1.2: Obama Social Events**
"President Obama comes out of the gate powerfully against childish republicans mccain graham", "This racist snowbilly was almost our vice president palin tweets that Obama is using shuck jive", "The republican party is done at the national level"	"First official Obama rally of 2012 in columbus ohio", "Exclusive interactive panoramic images from the last night of the democratic national convention", "There is nothing more powerful than ordinary citizens coming together for a just cause", "Michelle Obamas 2012 dnc speech with photos and transcript"
Com. 2: Obamacare	
"Obamacare signups Friday 970k or 1 3m private enrollments 3 3m total", "It's not your freedom youre worried about its theirs", "Debunking the republican lie that health insurance costs have skyrocketed under Obama", "Obamacare enrollment picks up as more Americans are getting covered", "Lowering budget with health insurance", "Republican Mike Shirkey working hard to solve an Obamacare problem that literally does not exist"	
Subcom. 2.1: Republicans	**Subcom. 2.2: Insurance**
"The president almost becomes his own anger translator over GOP Obamacare sabotage", "Republicans have now entered the hypocrisy zone on the Obamacare", "GOP Obamacare replacement shows conservatives have lost the health care battle"	"Funding for Michigan health centers will help the uninsured find coverage', "Obamacare day one my hideous experience", "More on GOP bill to require insurers to tell you how Obamacare is raising your rates the house republicans respond", "RNC fundraising email claims health insurance costs under Obama have gone up eight times higher than they have"

is a combination of subcommunities and our approach has the advantage of locating isolated documents which do not belong to any clusters. Additionally, the results presented closely resemble that of spectral clustering. We proved a connection between our algorithm and spectral clusterings by deriving the equivalence between the respective objective functions. We also provided a comparative study of the clustering algorithms on a data set with different densities and topologies and quantified the resluts via information theoretical metrics. Our studies show that community detection can be successfully used both for spatial- and for document clustering, opening the door for combining results from spectral clustering theory and other community detection methods in graph theory.

REFERENCES

[1] Atzmueller, M., Doerfel, S., Mitzlaff, F.: Description-oriented community detection using exhaustive subgroup discovery. Information Sciences 329, 965–984 (2016)
[2] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. the Journal of machine Learning research 3, 993–1022 (2003)
[3] Girvan, M., Newman, M.E.: Community structure in social and biological networks. Proceedings of the national academy of sciences 99(12), 7821–7826 (2002)
[4] Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. New Journal of Physics 11(3), 033015 (2009)
[5] Lancichinetti, A., Sirer, M.I., Wang, J.X., Acuna, D., Körding, K., Amaral, L.A.N.: High-reproducibility and high-accuracy method for automated topic classification. Physical Review X 5(1), 011007 (2015)
[6] Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: Proc. of ICML (2014)
[7] MacKay, D.J.: Information theory, inference and learning algorithms. Cambridge university press (2003)
[8] Manning, C.D., Raghavan, P., Schütze, H., et al.: Introduction to information retrieval, vol. 1. Cambridge university press Cambridge (2008)
[9] Martinsson, P.G., Rokhlin, V., Tygert, M.: A randomized algorithm for the decomposition of matrices. Applied and Computational Harmonic Analysis 30(1), 47–68 (2011)
[10] Shi, J., Malik, J.: Normalized cuts and image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22(8), 888–905 (2000)
[11] Von Luxburg, U.: A tutorial on spectral clustering. Statistics and computing 17(4), 395–416 (2007)

Third Party Effect: Community Based Spreading in Complex Networks

Christian Bauckhage, César Ojeda and Rafet Sifa
Fraunhofer IAIS
St. Ausgustin, Germany

Shubham Agarwal
Hewlett Packard Enterprise GmbH.
Böblingen, Germany

Christian Backhage and Rafet Sifa
University of Bonn
Bonn, Germany

***Abstract*—A substantial amount of network research has been devoted to the study of spreading processes and community detection without considering the role of communities in the characteristics of spreading processes. Here, we generalize the SIR model of epidemics by introducing a matrix of community infecting rates to capture the heterogeneous nature of the spreading as defined by the natural characteristics of communities. We find that the spreading capabilities of one community towards another is influenced by the internal behavior of third party communities. Our results provide insights into systems with rich information structure and into populations with diverse immunology responses.**

I. Introduction

Studying the dynamics of information propagation has become a major activity in the area of Web science. The nature of media such as e-mails, blogs, or social networks defines limits on how information spreads on the Web. For example, the spreading of tweets can occur within a time scale of minutes and hours [1]. On the other hand, information found in blogs can propagate within time scales of days or weeks [2]. In all these cases however, we can reduce the description of the communication patterns to a spreading process in a network which, among others, allows us to study when a particular news item becomes popular.

In this context, we study the interplay between the latent structure of the network of people partaking in a discussion, information spreading probabilities, and their impact on information epidemics. One defining aspect of network structure is the presence of *communities* or clusters of nodes which are commonly understood as sets of nodes that contain more connections among themselves than to the rest of the network [3]. Research in this direction aims at providing accurate and fast algorithms which unveil these communities. A key heuristic used in community detection algorithms is the concept of *homophily*, i.e. the tendency of nodes of similar characteristics to interconnect [4], [5].

Our goal in this paper is to study the properties of spreading processes where homophily affects not only the network structure but also spreading probabilities. It seems natural to assume that intrinsic characteristics of individual influence how they are connected to others and how they react to a particular spreading process. We study the well established *Susceptible-Infected-Recovered* (SIR) model [6], [7]. which represents the evolution of three states an agent can be in during an epidemic: An agent can be suceptible to an infection, can be infected, or may have developed an immune response and thus recovered from the particular infection.

Here we introduce a model that encodes the heterogeneous characteristics of this process in a matrix Λ where an arbitrary element $\lambda_{cc'}$ defines the rates of propagation from community c to community c'. We provide a novel algorithm to simulate the continuous time evolution of the spreading process and, using data from synthetic as well as empirical networks, we show that the heterogeneous nature of community modulated spreading plays a key role in hampering or supporting the spreading process.

Our model reveals hidden phenomena by incorporating the community structure which has not been considered in traditional models. In particular, we study the spreading process between two specific communities and show that the internal spreading of a third party community can noticeably impact the spreading process.

Previous work in this area studied the importance of community structure for spreading processes using the simpler *Susceptible-Infected-Susceptible* (SIS) model for epidemic outbreaks [8], [9]. The work in [8], [9] introduced algorithms to exploit the community structure in networks to detect vital nodes in the diffusion of diseases and thus allow for controlling the dynamics of an epidemic.It has been reported that the more defined the community structure is (i.e. the more connections exist within a community) the more likely an infection evolves in an isolated manner [10]. In all these related work however, the role of the community is merely seen from a structural perspective rather than explored as a factor that affects spreading processes. In other words, prior work in this area sees the diffusion of information as indifferent to the interests of a community. However, probabilistic models of community detection for social and consumer networks [3] indicate that the interest of a particular group is correlated with its connections. We thus aim at building a bridge between interests and infection rates.

In the context of information diffusion, this point of view illuminates the characteristics of the spreading a message itself. For example, in a social network, a piece of news regarding a novel algorithm may propagate faster in a community of computer scientists than in a community of physicists.

The remainder of the paper is organized as follows: first, we introduce details of our model, the algorithm required for its simulation, and the time alignment process to make use of

P. Haber et al. (Hrsg.), *Data Science – Analytics and Applications*, http://doi.org/10.1007/978-3-658-19287-7_5

Algorithm 1 Community Based SIR Algorithm

Input: Pre Infected nodes $I_0 = \{v\}$, Graph $\mathcal{G} \equiv (\mathcal{E}, \mathcal{V})$,
Input: Λ, Ω, ζ
Initialize I with all the pre-infected nodes.
Initialize Π for all the nodes of the network.
while $|I| > 0$ **do**
  if $\zeta = 0$ **then**
    $break$
  end if
  $\zeta = \zeta - 1$
  Extract node v from I
  for each $u \in \Gamma(v)$ **do**
    if $\Pi(u) = 1$ **then**
      $p \sim (0, 1)$
      if $p < \Lambda_{\sigma(v),\sigma(u)}$ **then**
        $\Pi(u) \leftarrow 0$
        Add node u to I
      end if
    end if
  end for
  $q \sim (0, 1)$
  if $q < \Omega(v)$ **then**
    $\Pi(v) \leftarrow 2$
  else
    Add node v to I
  end if
end while

the model in real life settings. Next, a hypothetical scenario of three communities is studied via synthetic networks to analyze different spreading scenarios. Finally, we use our model to simulate heterogeneous spreading process in real networks and conclude with a discussion and an outlook to future work.

II. A Community Based SIR Model

In this section, we formalize our notion of community based spreading. We begin by establishing the basic notation and algorithm and then discuss dynamical aspects of our approach, i.e. how to define the *physical time* of spreading processes through the Poisson process formalism. We represent a network as a graph $\mathcal{G} \equiv (\mathcal{E}, \mathcal{V})$ where $\mathcal{V}$ are the nodes and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ are the edges of the network. Each node of the network is assigned to one of C communities through the mapping $\sigma(v) \to c$ where $c \in [0, 1, ..., C]$ and $v \in \mathcal{V}$. In real world applications, this mapping is typically obtained through a clustering algorithm such as the Girswan-Newman modularity algorithm, among others. We introduce the spreading matrix $\Lambda = [\lambda_{c,c'}]$ with C rows and columns, $\Lambda \in \Re^{C \times C}$. This matrix encodes spreading probabilities among different communities of a network, that is, $\lambda_{c,c'}$ defines the spreading probability from community c to c'. In the following, we will refer to the diagonal and the off-diagonal elements of Λ as the intra- and inter-community spreading probabilities respectively. $\Gamma(v)$ defines the neighbors of the node *v* and $\Pi(v)$ tracks the node-state, which can be (S)usceptible, (I)nfected, or (R)ecovered. *I* defines a queue of infected nodes, initialized with the pre-infected nodes which are one of the inputs of our algorithm. We modify the algorithm in [7] to include community spreading. The algorithm works as follows: we extract a node *v* from *I* and then try to infect its susceptible neighbors $\Gamma(v)$. Spread to neighbor *u* is successful if a randomly generated number from a uniform distribution between 0 and 1, *p* is less than the spreading rate $\Lambda_{\sigma(v),\sigma(u)}$. When all susceptibles neighbors $\Gamma(v)$ have been checked for spreading, the node *v* is checked for recovery. The recovery rate $\Omega(v)$ of the node is compared with a newly generated random number *q*, again from a uniform distribution between 0 and 1. If the recovery is successful, $\Pi(v)$ is changed to the recovery state. Otherwise the node *v* is pushed back to the *I* queue. Whenever the state of a node changes, its state value is updated in Π. This procedure is repeated until there are no infected nodes anymore or the maximum number of cycles ζ is reached (see Alg. 1).

In order to properly study the dynamical behavior of spreading processes, we differentiate between algorithmic and continuous time. Algorithmic time is defined by the number of cycles which update the different states during the course of a simulation. Continuous time, however, is defined by the different probabilities of events, encoded in the infection matrix and recovery probabilities. It accounts for the *real physical time of a simulation.*

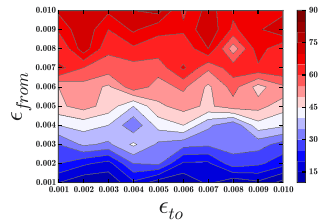
For instance, suppose that only three nodes are infected and each of them posses only one neighboring node, all of them different and susceptible. This is a simple network of 6 nodes and tree edges. Thinking in physical time, the first node might infect its neighbor during the morning, the second node might infects it neighbor at noon, and the last node might infect its neighbor during the afternoon. The algorithmic cycle runs through three different test cycles, but the real time is defined trough the inter event times between infections. Since each occur independently and each can be though as a Poisson process (*an infection of a node*), the inter event rate of the joint process, (*an infection in the network*) can be defined by adding the rates of each infection[1]. For every change of state during the course of a simulation, we must thus record the sum of the rates of all possible processes, i.e. the sum of all infection rates for edges with one node infected and one susceptible, as well as all the recovery rates. The time elapsed between two given events will be obtained by sampling from an exponential distribution using this sum as a rate. This updating process is incorporated into Alg. 1 and applied every time an infection or a recovery occurs to keep track of the time of the infection. This approach is know as the kinetic monte carlo method and is widely used in physics [11].

III. Analyzing a Synthetic Network

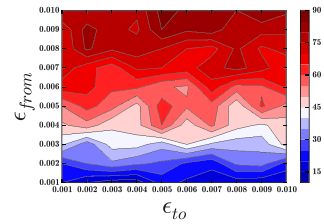
Since the main concern of this paper is to study the interaction among different communities in the context of heterogeneous spreading processes, we focus our study on two major aspects:

- **Intra-community Spreading:** Describes spreading processes within communities, which are governed by the diagonal elements of the spreading matrix $\lambda_{c,c}$

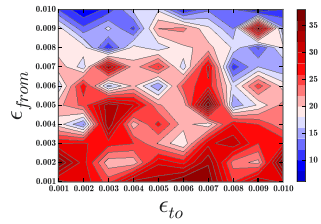
[1]The rates r are obtained from a probabilities p in Λ or Ω via the transformation $r = -\log(1 - p)$



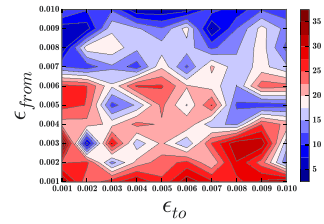
(a) Epo vs ϵ_{to} and ϵ_{from} for $\lambda_{2,2} = 0.1$.
Lower intra communiy spreading rate for third party



(b) EPo vs ϵ_{to} and ϵ_{from} for $\lambda_{2,2} = 0.1$,
Higher intra community rate for third party



(c) $\mid D_{\mathbf{C0}} - D_{\mathbf{C1}} \mid$ vs ϵ_{to} and ϵ_{from} for $\lambda_{2,2} = 0.1$,
Lower intra communiy spreading rate for third party



(d) $\mid D_{\mathbf{C0}} - D_{\mathbf{C1}} \mid$ vs ϵ_{to} and ϵ_{from} for $\lambda_{2,2} = 0.8$,
Higher intra community rate for third party

Fig. 1: Phase diagram $|D_2 - D_1|$ for a synthetic network of 3 communities with 500 nodes per community. Each community posses 15 edges to other communities selected at random, for an average number of 30 outer edges. We perform simulations for different values of the third party intra community spreading $\lambda_{2,2}$.

- **Inter-community Spreading:** Describes spreading processes across distinct communities, which are governed by the off diagonal elements of the spreading matrix $\lambda_{c,c'}$

That is, we aim to dissect how the interplay between two communities, which is essentially an inter-community process, is affected by intra-community interactions. To analyze this effect, we consider a hypothetical scenario in which interactions among three different communities **C0**, **C1**, **C2** is examined. We use a network of 500 nodes per community. Each community is generated as a complex network following a power law degree distribution $P(k) \sim k^{-3}$. Each community has 15 edges to other communities selected at random, for an average number of 30 outer edges. The **C0** community is defined as the community in which a spreading process starts, i.e. the initially infected nodes in *I* only belong to this community. Community **C1** is the target community where we quantify the impact of the infection in this community via proxies defined below. Finally, **C2** is the third party community whose inter-community spreading will come under scrutiny by measuring the produced impact (number of infected nodes) and the temporal behavior of the peaks of **C0** and **C1**.

A. Impact Analysis

In order to quantify the impact of the source community **C0** on the target community **C1**, we define the epidemic potential of infection of a community (*EPo*) [12]. We calculate the *EPo* value by performing 100 simulations with a given density of initially infected nodes in **C0** and then counting the number of simulations that cause at least half of the nodes in **C1** to be infected. This way, we measure how likely it is for a random node in the target community to be infected by an infection starting in the source community. Although the initial density in the source community is kept fixed, each simulation starts with a different random set of nodes being infected. To understand the role of the inter community spreading, we study the *variation* of two different parameters:

- The infection rate towards the *source* community is defined by ϵ_{to} such that $\epsilon_{to} = \lambda_{c,0}$ where $c \neq 0$
- The infection rate from the *source* community is defined by ϵ_{from} such that $\epsilon_{from} = \lambda_{0,c'}$ where $c' \neq 0$.

The higher the value of *Epo*, the higher the impact value is and hence a larger number of nodes are likely to be infected in the target community. The phase diagrams in Fig. 1 represent the epidemic impact score for different ϵ_{to} and ϵ_{from}; impact scores are plotted as heatmaps. If the impact score for a given ϵ_{to} and ϵ_{from} is low, the corresponding point for ϵ_{to} and ϵ_{from} in the phase diagram is represented with a cold blue color. Similarly, if the *EPo* is higher, the corresponding point in the phase diagram is shown in a a hot red color. ϵ_{to} and ϵ_{from} are represented along the X-axis and Y-axis, respectively. Whether a given value is high or low is defined relative to the maximum and minimum value obtained in the analysis of the full parameter space.

We perform simulations for two different values of the third party intra community spreading $\lambda_{2,2}$. In Figs. 1a and 1b, we

have *EPo* values of over 60 for values of the infection rate from the source ϵ_{from} between 0.006 and 0.01 . On the other hand for ϵ_{from} values between 0.001 and 0.004 we have low values of less than 30 *EPo*. This is expected since we are studying the infections rates from the source. If we increase ϵ_{from} we increase the likelihood of an infection to develop in the target. On the contrary, if we study the changes along the ϵ_{to} axis almost no changes in the value of the *EPo* are observed. In this case, the role of infections towards the source community is minor. The ϵ_{to} parameters only plays a role in *EPo* of the target trough a spreading path starting in the source, going to an outer community and returning to the source before reaching the target. Since we are in the present of complex networks, many paths within the community will occur where hubs (highly interconnected nodes) come into play. Hence any added advantage from the outside of the community is neglected.

If we compare Fig. 1a and Fig. 1b we find the main result of this paper. The *EPo* value over a target community is influenced by the intra community behavior of third parties. For $\lambda_{2,2} = 0.1$ (Fig. 1a) we have *Epo* over 45 for ϵ_{from} above 0.005. For $\lambda_{2,2} = 0.9$ (Fig. 1b) we have *Epo* values over 45 for ϵ_{from} above 0.004. The stronger a spreading occurs in some other part of the network, the more likely it is that the target gets infected. In other words, the source is able to use the third party as an enhancer of its spreading capabilities. *This indicates a rather counterintuitive result, the optimal strategy for spreading information to a target might as well be to increase the spreading of a third party.*

B. Dynamical Analysis

In order to characterize the dynamical behavior of an epidemic under the heterogeneous framework, we study the time difference between infection peaks. We show an example of a simulation in our hypothetical network of tree communities in Fig. 2 which plots the number of infected nodes $I_c(t)$ at time t for each community. Note that the target community **C1** shows a infection profile which is delayed by 30 time units compared to the source **C0**. We define the average peak delay *APd* as $\langle \mid D_{\mathbf{C0}} - D_{\mathbf{C1}} \mid \rangle$ where D_c is the time at which the maximum amount of nodes are infected in community c. We performed 100 different simulations and obtained the absolute value of the difference $\mid D_{\mathbf{C0}} - D_{\mathbf{C1}} \mid$ and the average over the simulations. In a similar fashion as in Fig. 1a and Fig. 1b we study the phase diagram for *APd* between the source and the impact community. Using the same networks and parameter as above, we obtained the ϵ_{to} and ϵ_{from} phase diagram for two different values of the intra community spreading of the third party $\lambda_{2,2}$. We present the results in Fig. 1c and Fig. 1d. As before we observe two different regions. For Fig. 1c ($\lambda_{2,2} = 0.1$) we have low $\langle \mid D_{\mathbf{C0}} - D_{\mathbf{C1}} \mid \rangle$ values above $\epsilon_{from} = 0.008$ whereas for Fig. 1d ($\lambda_{2,2} = 0.9$) we have low values above $\epsilon_{from} = 0.007$. Low values of the peaks occur when the spreading process spreads quickly between communities. In this case, both infection processes will happen almost simultaneously and peaks will be attained

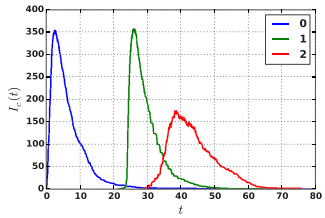


Fig. 2: Example of an infection profile in real time with $\epsilon_{to} = 0.008$ and $\epsilon_{from} = 0.004$, $\lambda_{2,2} = 0.1$ showing the number of infected nodes for tree communities **C0**, **C1** and **C2** that form a synthetic network. See text for details.

at similar times. This scenario is given for large values of ϵ_{from}. Clearly, once the third party community increases its intra community interaction, it increases the chances of the target community to get infected trough the third party. Consequently, we observe a lower value of ϵ_{from} for the beginning of the low $\langle \mid D_{\mathbf{C0}} - D_{\mathbf{C1}} \mid \rangle$ region in Fig. 1d.

In contrast to the *EPo* diagrams we have higher noise fluctuations. We can find low values of the *Apd* in the high value region. This occurs as a consequence of the complex nature of the network. The delay between the peaks show strong changes depending on when the infection process encounters hubs or highly connected nodes. This creates a bigger standard deviation for the $\langle \mid D_{\mathbf{C0}} - D_{\mathbf{C1}} \mid \rangle$ values (*APd*) and, in consequence, more fluctuations.

IV. Analyzing Real Networks

We now explore the behavior of the heterogeneous process model in more complex structures. For that, we study community based spreadings in real world networks.

We work with the SNAP data set [13] which contains email communication networks, peer to peer communication networks and collaboration networks, defined by co-authorship patterns of researchers. The diversity of these networks provides examples of the application potential of our work in different information diffusion contexts.

To detect communities we used the Newman-Girwan algorithm for community detection [14]. This algorithm yields a hard clustering of the nodes of a network and has the added advantage of automatically detecting the appropriate number of communities. We show the results of our approach in Table I. It indicates the number of nodes and edges for each network as well as the number of communities found. We define the source, target and third party as the tree biggest communities in each network. The spreading rate above which a network ensures an epidemic (*epidemic threshold*) is largely dependent on the nature of the network and its adjacency matrix. Consequently, each network will require different rate values to guarantee that an epidemic spreads. We thus start with a inter community rate of $\lambda_{c,c' \neq c} = 0.1$ and intra community values for the rest of the network of $\lambda_{c,c} = 0.2$. When

Name	# Communities	# Edges	# Nodes	*Epo*(1)	*Epo*(2)	$N(C_1)/N(C_0)$	$N(C_2)/N(C_0)$
CA-GrQc	421	14484	5242	2	10	0.87	0.36
CA-HepPh	416	118489	12008	98	98	0.92	0.32
CA-HepTh	546	25973	9877	14	14	0.95	0.93
Enron	1589	183831	36692	24	26	0.81	0.48
p2p-Gnutella04	25	39995	10876	6.67	13.33	0.96	0.78
p2p-Gnutella05	22	31840	8846	73.33	100.00	0.99	0.76
p2p-Gnutella06	20	31526	8717	26.67	60.00	0.83	0.77
p2p-Gnutella08	22	20778	6301	40.00	60.00	0.95	0.70
p2p-Gnutella09	30	26014	8114	20.00	73.33	0.89	0.78
p2p-Gnutella24	50	65370	26518	26.67	20.00	0.98	0.72
p2p-Gnutella25	58	54706	22687	6.67	6.67	0.86	0.83

TABLE I: Results on real world networks from the SNAP data set [13]. $N(C_i)/N(C_0)$ is the ratio of sizes of the target (third party) community to the source community. For source and target community we used an intra community spreading of $\lambda_{c,c} = 0.1$. (1) Stands for a third party intra community spreading of $\lambda_{2,2} = 0.2$ and (2) $\lambda_{2,2} = 0.8$.

we do not observe any impact we increase these values by 0.05 and repeat the simulation. Additionally, Table I includes the *EPo* values for the target with *EPo*(1) with $\lambda_{2,2} = 0.1$ and *EPo*(2) $\lambda_{2,2} = 0.8$. All simulation start with 1% of randomly chosen nodes in the source infected. We also show the ratio of the community sizes $N(C_i)$ is the number of nodes in community C_i. As opposed to the hypothetical scenario of last section we have no guarantee that there exist any connection at all between the tree communities **C0**, **C1**, **C2**. In most cases, we can once again see the important role played by the third party. In some cases, we are able to observe a 30% increase in the value of the *EPo*.

V. Conclusions and Future Work

In this paper, we studied the heterogeneous nature of (information) spreading processes in networks. In particular, we aimed at answering the question: do the internal behaviors of third parties affect the interaction between two communities? We extended the **SIR** framework to include particular parameters for the behavior of network communities and studied spreading in a hypothetical network of tree communities. Dynamical phase diagrams showed that third parties do indeed affect the spreading between two communities. We also applied our model to real world networks and found that the enhancing effect through third parties can also be observed in networks with complex community structure.

Our results indicate that, if we want to start a spreading process from one community to another, we should study how likely this spreading process will reach other communities as well as the desired target. The optimal strategy should include the added advantage of the communication patterns of other communities. We use the third party as an amplifier of the spreading capabilities of the source. These results are only visible through the study of heterogeneous spreading processes since the different effect of the third parties is captured only in this context.

In future work, we aim at studying the role of other parameters of the networks such as the inter-community connectivity, and community sizes. We need to explore how these network characteristics modify the role of the third party effect. Also, new centrality measures are required which incorporate the heterogeneous nature of the spreading. The importance of a node will be dependent on its connections with communities that have high spreading rate.

References

[1] I. Taxidou, A. Alzoghbi, P. M. Fischer, and C. Schöller, "Towards real-time lifetime prediction of information diffusion," in *Proc. of ACM Web Science*, 2015.

[2] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *Proc. of ACM SIGKDD*, 2009.

[3] J. Yang and J. Leskovec, "Overlapping community detection at scale: a nonnegative matrix factorization approach," in *Proc. of ACM WSDM*, 2013.

[4] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.

[5] F. Shah and G. R. Sukthankar, "Using network structure to identify groups in virtual worlds." in *Proc. of ICWSM*, 2011.

[6] R. M. Anderson, R. M. May, and B. Anderson, *Infectious diseases of humans: dynamics and control*. Wiley Online Library, 1992, vol. 28.

[7] N. Antulov-Fantulin, A. Lančić, H. Štefančić, and M. Šikić, "FastSIR algorithm: A fast algorithm for the simulation of the epidemic spread in large networks by using the susceptible–infected–recovered compartment model," *Elsevier Information Sciences*, vol. 239, pp. 226–240, 2013.

[8] X. Wu and Z. Liu, "How community structure influences epidemic spread in social networks," *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 2, pp. 623–630, 2008.

[9] W. Huang and C. Li, "Epidemic spreading in scale-free networks with community structure," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2007, no. 01, p. P01014, 2007.

[10] M. Salathé and J. H. Jones, "Dynamics and control of diseases in networks with community structure," *PLoS Comput Biol*, vol. 6, no. 4, p. e1000736, 2010.

[11] A. F. Voter, "Introduction to the kinetic monte carlo method," in *Radiation Effects in Solids*. Springer, 2007, pp. 1–23.

[12] G. Lawyer, "Understanding the Influence of All Nodes in A Network," *Nature Scientific Reports*, vol. 5, 2015.

[13] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, jun 2014.

[14] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

Cosine Approximate Nearest Neighbors

David C. Anastasiu
Department of Computer Engineering
San José State University, San José, CA, USA

***Abstract*—Cosine similarity graph construction, or all-pairs similarity search, is an important kernel in many data mining and machine learning methods. Building the graph is a difficult task. Up to n^2 pairs of objects should be naïvely compared to solve the problem for a set of n objects. For large object sets, approximate solutions for this problem have been proposed that address the complexity of the task by retrieving most, but not necessarily all, of the nearest neighbors. We propose a novel approximate graph construction method that leverages properties of the object vectors to effectively select few comparison candidates, those that are likely to be neighbors. Furthermore, our method leverages filtering strategies recently developed for exact methods to quickly eliminate unpromising comparison candidates, leading to few overall similarity computations and increased efficiency. We compare our method against several state-of-the-art approximate and exact baselines on six real-world datasets. Our results show that our approach provides a good tradeoff between efficiency and effectiveness, showing up to 35.81x efficiency improvement over the best alternative at 0.9 recall.**

***Index Terms*—Cosine, all-pairs similarity search, nearest neighbors, graph construction, similarity graph.**

I. Introduction

Cosine similarity graph construction, or all-pairs similarity search (APSS), is an important kernel in many data mining and machine learning methods, including ones for pattern recognition [1], online advertising [2], query refinement [3], and collaborative filtering [4]. The goal of APSS is to find, for each object in a set, which is usually referred to as a *query*, all other objects that are sufficiently similar, i.e., those with similarity of at least some threshold ϵ. In this work, we focus on objects that are represented as *non-negative sparse vectors*, which applies to many real-world objects. For example, a social network graph is often represented through its adjacency matrix, where a row encodes the neighborhood of a user and non-negative feature weights are assigned to describe the closeness of the relationship between the user and all other users in the network. Similarly, a book in a library can be represented using the bag-of-words model by a vector of word frequencies.

One way to construct the similarity graph is to compare each query object against all other objects, which we call *candidates*, and filter out those that have similarity below ϵ. However, this will require $n(n-1)/2$ similarity computations and does not scale to large sets of objects. Moreover, many of the candidates will likely be filtered, yet this naïve approach still computes their similarities to the query. A number of methods have been developed in the last decade to reduce the set of candidates. Chaudhuri et al. [5], for example, found that only some of the leading features in each vector (which they call the *prefix* of the vector) had to be considered to find all potential candidates. In other words, if a candidate does not have any non-zero value for a feature in the set of query prefix features, the similarity of that candidate with the query will necessarily be below ϵ and the candidate can be ignored, or pruned. Bayardo et al. [3] used this idea to develop an exact APSS method, AllPairs, which has since been extended by several researchers. In a previous work [6], we gave an overview of these extensions and provided exact and approximate cosine APSS algorithms, L2AP and L2AP-Approx, that significantly outperformed previous methods.

A popular approach for approximate nearest neighbor search has been locality sensitive hashing (LSH). LSH first constructs a search data structure by using families of locality sensitive hash functions, which map similar objects to the same bucket with high probability, to place each object in one or more buckets. Then, at query time, the objects in the buckets that the query maps to will be the candidate set that will be compared with the query. The generic LSH data structures have been found to perform poorly for the APSS problem (see [3], [6]) when expecting high average recall, due to large candidate sets that must be compared with the query. However, Satuluri and Parthasarathy [7] developed a principled Bayesian algorithm (BayesLSH) that uses LSH based estimates to prune away a large majority of the false positive candidates.

Solving a related problem, the k-nearest neighbor graph (kNNG) construction problem, where we are interested in the k objects with the highest similarity to each query, Park et al. [8] discovered that objects with shared high weight features are more likely to be neighbors. Additionally, Dong et al. [9] showed that additional neighbors may be found by considering a neighbor's neighbors as candidates. We combined these ideas [10] to design an approximate kNNG construction method which was used as the first step in an exact kNNG solution. The neighbor similarity concept was also independently used by Malkov et al. [11] to develop a greedy approximate kNNG construction algorithm that works by traversing a small-world neighborhood graph from a random start node.

In this paper, we describe a novel approximate APSS method that leverages properties of the object vectors and their neighbors to effectively select few comparison candidates. Our method works in two steps. First, we leverage the prefix filtering and feature weight priority ideas to quickly construct an approximate $\min -\epsilon$ kNNG, while ignoring unsuitable candidates. In the second step, our method traverses the graph to identify candidates, prioritizing those objects that are likely to be a part of the APSS solution. Unlike the naïve approach, the complexity of our method is much smaller than $O(n^2)$, yet it leads to identifying most of the nearest neighbors up to 35.81x faster than the best alternative at 0.9 recall. Our contributions are as follows:

- We propose `CANN`, a novel approximate algorithm for solving the APSS problem. Unlike previous methods, which use filtering based candidate generation, `CANN`

P. Haber et al. (Hrsg.), *Data Science – Analytics and Applications*, http://doi.org/10.1007/978-3-658-19287-7_6

uses feature and neighborhood graph weights to gather a small set of favorable candidates that will be compared with a query.
- We analyze the properties of neighborhood graphs of six real-world datasets at varying minimum similarity thresholds and draw conclusions on the utility of the APSS output for solving data mining problems.
- We conduct extensive experiments that measure both the effectiveness and efficiency of our method, compared to several state-of-the-art approximate and exact APSS baselines.

The remainder of the paper is organized as follows. We give a formal problem statement and describe our notation in Section II. In Section III, we present our algorithm. In Section IV, we describe the datasets, baseline algorithms, and performance measures used in our experiments. We present our experiment results and discuss their implications in Section V, and Section VI concludes the paper.

II. Problem statement

Given object d_i in a set D of n objects, we seek to find its nearest neighbors, the set of objects in $D \setminus \{d_i\}$ whose cosine similarity with d_i is at least ϵ. The ϵNNG of D is a directed graph $G = (V, E)$ in which vertices correspond to the objects and an edge (v_i, v_j) indicates that the jth object is among the nearest neighbors of the ith object, i.e., their similarity is at least ϵ. An approximate ϵNNG may not contain all the nearest neighbors for each object, yet the present edges will denote actual neighbors. In other words, we assume the similarity between objects is computed exactly and those objects with similarity below ϵ are not considered neighbors.

Borrowing the notation from [10], we will use d_i to indicate the ith object, $\mathbf{d}_i$ to indicate the feature vector associated with the ith object, and $d_{i,j}$ to indicate the value (or weight) of the jth feature of object d_i. Since the cosine function is invariant to changes in vector lengths, we assume that all vectors have been scaled to be of unit length ($||\mathbf{d}_i|| = 1, \forall d_i \in D$), which simplifies computing the similarity between two vectors $\mathbf{d}_i$ and $\mathbf{d}_j$ to their dot-product, which we denote by $\langle \mathbf{d}_i, \mathbf{d}_j \rangle$.

An *inverted index* is a data structure that has been extensively used in Information Retrieval and Data Mining to speed up similarity computations for sparse data. It consists of a set of m lists, $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$, one for each feature, such that list I_j contains pairs $(d_i, d_{i,j})$, where d_i is an indexed object with a non-zero weight for feature j and $d_{i,j}$ is that weight.

Given some dimension p, the *prefix* (vector) $\mathbf{d}_i^{\leq p}$ can be thought of as the same vector $\mathbf{d}_i$ with all values for features $j, j > p$, set to 0. The *suffix* vector $\mathbf{d}_i^{>p}$ is analogously defined. Given these definitions, it is easy to verify that the dot-product of a query vector $\mathbf{d}_q$ with a candidate $\mathbf{d}_c$ can be decomposed as the sum of the candidate prefix and suffix dot-products with the query,

$$\langle \mathbf{d}_q, \mathbf{d}_c \rangle = \langle \mathbf{d}_q, \mathbf{d}_c^{\leq p} \rangle + \langle \mathbf{d}_q, \mathbf{d}_c^{>p} \rangle.$$

III. Constructing the similarity graph

In this section, we will describe our approximate APSS method, CANN. Our method works in two steps, as shown in Algorithm 1. First, CANN leverages features with high weight in object vectors to quickly construct an initial approximate min $-\epsilon$ kNNG. Unlike previous kNNG construction methods, CANN eliminates from consideration the majority of the objects that cannot have a similarity of at least ϵ when building the graph. Limiting the neighborhood size to k allows our method to construct an initial similarity graph while bounding the overall memory usage. In the second step, our method traverses the initial graph to identify candidates for building the ϵNNG, prioritizing objects that are likely to be in the APSS solution. The following subsections discuss the steps in detail.

Algorithm 1 The CANN algorithm

```
1: function CANN (D, ϵ, k, μ1, μ2)
2:     N ← InitialGraph (D, ϵ, k, μ1)
3:     ImproveGraph (D, ϵ, k, μ2, N)
```

A. Min-ϵ kNNG construction

The initial graph construction step is detailed in Algorithm 2. CANN first builds a partial inverted index for the objects, indexing only a few of the leading features in each vector. Then, it uses a sorted version of the inverted lists and of the vectors to prioritize candidates that are selected for comparison with each query.

When choosing candidates to be compared with a query object, CANN takes advantage of the prefix filtering idea [5] as a way to automatically eliminate many objects that cannot be similar enough. As applied in our method, it states that a query object cannot be similar enough if it has no features in common with a candidate in its prefix, given an appropriately chosen prefix. There are several ways to select the prefix features for a vector that have been detailed in the literature [3], [5], [6]. We use a simple method, which we first proposed in [6], that is both effective and efficient to compute. CANN indexes the leading features of each vector until the L2-norm of its suffix falls below the threshold ϵ.

To see why prefix filtering is useful, consider a candidate d_c that has prefix features $\{2, 7, 11\}$ and suffix features $\{12, 15, 19\}$, and a query d_q that has non-zero features $\{1, 9, 12, 15, 19\}$. According to the prefix selection principle, the suffix norm of d_c, $\|\mathbf{d}_c^{>j}\| < \epsilon$. Additionally, note that $\|\mathbf{d}_q\| = 1$, since all vectors are normalized. Therefore, even though the query and candidate have many features in common in the candidate suffix, based on the Cauchy-Schwarz inequality, their suffix dot-product will be below ϵ,

$$\langle \mathbf{d}_q, \mathbf{d}_c^{>p} \rangle \leq \|\mathbf{d}_q\| \|\mathbf{d}_c^{>p}\| < \epsilon.$$

Since the query does not have any non-zero values for the $\{2, 7, 11\}$ features, thus it has no features in common with the candidate in its prefix, the prefix dot-product $\langle \mathbf{d}_q, \mathbf{d}_c^{\leq p} \rangle$ will be 0, which means the overall similarity of the vectors will be below ϵ. CANN automatically avoids this object by only choosing candidates for the query d_q from the posting lists in the partial inverted index associated with non-zero features in the query.

CANN chooses up to μ_1 candidates to compare with each query by iterating through objects in inverted index lists. Based on the idea that objects with high weight features in common with the query are more likely to be neighbors [10], our method prioritizes candidates in each list by first sorting the lists in non-increasing value order. Moreover, CANN chooses list processing order based on the non-increasing query weight value of their associated features. Taking advantage of the commutativity property of cosine similarity, CANN only chooses candidates that follow the query in the

Algorithm 2 Min-ϵ kNNG construction in CANN

function INITIALGRAPH(D, ϵ, k, μ_1)
  $N_i \leftarrow \emptyset$ **for** $i = 1, \ldots, n$ ▷ Neighbor lists
  $L \leftarrow \emptyset$ ▷ Candidate list
  $T \leftarrow \emptyset$ ▷ Processed items
  $H \leftarrow \emptyset$ ▷ Query hash table
  for each $q = 1, \ldots, n$ **do** ▷ Create partial inverted index
    for each $j = 1, \ldots, m$ s.t. $d_{q,j} > 0$ and $\|\mathbf{d}_q^{>j}\| \geq \epsilon$ **do**
      $I_j \leftarrow I_j \cup \{(d_q, d_{q,j})\}$
  Sort inverted lists in non-increasing value order.
  for each $q = 1, \ldots, n$ **do**
    $T[d_c] \leftarrow 1$ **for all** $(d_c, s) \in N_q$; $l \leftarrow 0$
    for each $(j, q_j) \in sorted(\mathbf{d}_q)$ **do** ▷ non-increasing value order
      for each $(d_c, d_{c,j}) \in I_j$ **while** $l < \mu_1$ **do**
        if $d_c > d_q$ **and not** $T[d_c]$ **then**
          $s \leftarrow BoundedSim(d_q, d_c, \epsilon)$
          if $s \geq \epsilon$ **then**
            $L \leftarrow L \cup (d_c, s)$
            if $|N_c| < k$ **then**
              $N_c \leftarrow N_c \cup (d_q, s)$
          $T[d_c] \leftarrow 1$
        $l \leftarrow l + 1$
    Add current neighbors from N_q to L.
    $N_q \leftarrow$ neighbors with top-k similarity values in L.
  return $\bigcup_{i=1}^{n} N_i$

processing order. Additionally, it avoids comparing an object multiple times with the query by tagging it as done (1) in a bit-vector data structure (T). When the computed similarity is above ϵ, the candidate is added to the list L, and the query is added to the candidate's neighborhood if its size is below k.

Algorithm 3 Bounded similarity computation with pruning

function BOUNDEDSIM(d_q, d_c, ϵ)
  $s \leftarrow 0$
  for each $j = 1, \ldots, m$ **s.t.** $d_{c,j} > 0$ **do**
    if $d_{q,j} > 0$ **then**
      $s \leftarrow s + d_{q,j} \times d_{c,j}$
      if $s + \|\mathbf{d}_q^{>j}\| \times \|\mathbf{d}_c^{>j}\| < \epsilon$ **then**
        return -1
  return s

Since each query will be compared against many candidates, CANN uses a hash table to store non-zero query features and their associated suffix norms. This allows dot-products to be computed in a similar way to a sparse-dense vector dot-product (Algorithm 3), iterating only through the non-zero values of the candidate and looking up query values in the hash table[1]. However, CANN does not fully compute the dot-product in most cases. After each successful multiply-add, it computes an upper-bound estimate on the similarity based on the Cauchy-Schwarz inequality applied to the query and candidate suffix vectors. If what has been computed thus far (s), which amounts to the prefix dot-product of the vectors, plus the suffix similarity estimate is below ϵ, the computation can be safely terminated and the candidate pruned.

B. Candidate selection for similarity search

In its second step, CANN finds the nearest neighbors for each query by traversing the initial kNNG. It first creates an inverted index of the graph's adjacency matrix, which helps avoid re-computing similarities for candidates in whose neighborhoods the query already resides (reverse neighborhood). After tagging both objects in the query's neighborhood and reverse neighborhood, CANN uses a max heap to prioritize neighborhoods it should traverse in search for candidates, namely the neighborhoods of neighbors with high similarity. From those neighborhoods, CANN will pick up to μ_2 candidates, in non-increasing order of their similarity values. Those candidates who are not pruned by the *BoundedSim* function are included in the output. Finally, CANN updates the stored top-k list of neighbors as a way to improve the search for subsequent queries.

[1]We omitted the hashing from the algorithms to simplify the presentation.

Algorithm 4 ϵNNG construction in CANN

function IMPROVEGRAPH(D, ϵ, k, μ_2, N)
  $I \leftarrow Index(N)$ ▷ Graph adjacency matrix inverted index
  for each $q = 1, \ldots, n$ **do**
    $L \leftarrow \emptyset$; $T \leftarrow \emptyset$; $H \leftarrow \emptyset$; $l \leftarrow 0$
    $Q \leftarrow \emptyset$ ▷ Max heap
    $T[d_c] \leftarrow 1$ **and** $L \leftarrow L \cup (d_c, s)$ **for all** $(d_c, s) \in I_q$
    $T[d_c] \leftarrow 1$ **for all** $(d_c, s) \in N_q$
    $Insert(Q, (d_c, s))$ **for all** $(d_c, s) \in N_q$
    while $Size(Q) > 0$ **do**
      $(d_c, s) \leftarrow Extract(Q)$
      $L \leftarrow L \cup (d_c, s)$
      if $l < \mu_2$ **then**
        for each $(d_b, v) \in N_c$ **do**
          if not $T[d_c]$ **then**
            $s \leftarrow BoundedSim(d_q, d_c, \epsilon)$
            if $s \geq \epsilon$ **then**
              $Insert(Q, (d_b, s))$
            $T[d_c] \leftarrow 1$
      $l \leftarrow l + 1$
    Output d_q neighbors in L.
    $N_q \leftarrow$ neighbors with top-k similarity values in L.

C. Complexity analysis

CANN pre-computes and stores suffix L2-norms for all non-zero features in the vectors, which takes $O(z)$ time, where z is the number of non-zeros. The pre-processing and sorting steps are overshadowed by the similarity computations. CANN computes at most $n \times \mu_1$ similarities in step 1 and $n \times \mu_2$ similarities in step 2. Therefore, the overall complexity of CANN is $O(n(\mu_1 + \mu_2)\nu) \ll O(n^2\nu)$, where ν is the average number of non-zeros in a dataset vector.

IV. Experiment setup

A. Methods

We compare our approach, CANN, against our previous exact method, L2AP [6], and two approximate APSS variants, L2AP-Approx [6] (which we denote by L2AP-a) and BayesLSH-Lite [7] (denoted as BLSH-l). Efficient C/C++ based implementations for the baselines were made available by their respective authors. Unlike our method, since it seeks the exact solution, L2AP generates all potential candidates, not just those likely to be in the ϵNNG. It uses a slightly more complex way to determine vector prefixes than our method, but uses the same L2-norm pruning strategy when generating candidates. L2AP uses many different filtering conditions to prune the majority of false-positive candidates, which results in efficient exact ϵNNG construction. BLSH-l uses the same candidate generation strategy as AllPairs [3], but prunes many of the candidates through similarity estimates obtained through Bayesian inference. L2AP-a combines the candidate generation step from L2AP with some of the L2AP verification strategies and the Bayesian pruning in BLSH-l.

B. Datasets

We consider six real-world datasets in our work, with were graciously provided by Venu Satuluri and were also used in [7] and [6]. They represent three text collections (RCV1, WW500k, and WW100k), and three social networks

(Twitter, Wiki, Orkut), whose statistics are provided in Table I. Both link and text-based datasets are represented as TF-IDF weighted vectors. We present additional details below.

- **RCV1** is a standard benchmark corpus containing over 800,000 newswire stories provided by Reuters, Ltd. for research purposes, made available by Lewis et al. [12].
- **WikiWords500k** was kindly provided to the authors by Satuluri and Parthasarathy [7], along with the WikiWords100k and WikiLinks datasets. It contains documents with at least 200 distinct features, extracted from the September 2010 article dump of the English Wikipedia[2] (Wiki dump).
- **WikiWords100k** contains documents from the Wiki dump with at least 500 distinct features.
- **TwitterLinks**, first provided by Kwak et al. [13], contains *follow* relationships of a subset of Twitter users that follow at least 1,000 other users. Vectors represent users, and features are users they follow.
- **WikiLinks** represents a directed graph of hyperlinks between Wikipedia articles in the Wiki dump.
- **OrkutLinks** contains the friendship network of over 3M users of the Orkut social media site, made available by Mislove et al. [14]. Vectors represent users, and features are friends of the users. A user could have at most 1000 friends in Orkut.

TABLE I
DATASET STATISTICS

Dataset	n	m	nnz
RCV1	804,414	43,001	61M
WW500k	494,244	343,622	197M
WW100k	100,528	339,944	79M
Twitter	146,170	143,469	200M
Wiki	1,815,914	1,648,879	44M
Orkut	3,072,626	3,072,441	223M

For each dataset, n is the number of vectors/objects (rows), m is the number of features (columns), and nnz is the number of non-zeros.

C. Execution environment and evaluation measures

Our method and all baselines are serial programs. CANN was implemented in C and compiled with gcc 6.0.1 (-O3 enabled). Each method was executed, without other running programs, on a server with dual-socket 2.8 GHz Intel Xeon X5560 (Nehalem) processors and 24 Gb RAM. We varied ϵ between 0.3 and 0.9, in increments of 0.1. We measure efficiency as the total execution time for the method (wall-clock, in seconds).

We tested each approximate method under a large range of meta-parameters. We tested BLSH-l and L2AP-a by setting the expected false negative rate (ϵ in [7]) to each value in $\{0.01, 0.02, \ldots, 0.09, 0.1, 0.2, \ldots, 1.0\}$. We also varied the number of hashes, h, testing values in $\{128, 256, 384, 512\}$. We tested CANN with k in $\{1, 5, 10, 25, 50, \ldots, 250\}$ and set $\mu_1 = \alpha \times k$, given $alpha \in \{1, 2, \ldots, 10\}$, and $\mu_2 = 5 \times \mu_1$. For each combination of parameters, we executed the method three times and averaged the resulting execution times. We report, at each level of recall, the best execution time for the method given our manual parameter search.

We use average recall to measure the accuracy of the constructed ϵNNG. We obtain the correct ϵNNG via a brute-force search, then compute the average recall as the mean of recall values for each query, were recall is computed as the fraction of (all) relevant/true neighbors that were included by the algorithm in the query's neighborhood.

[2]http://download.wikimedia.org

V. RESULTS & DISCUSSION

Our experiment results are organized along several directions. First, we analyze statistics of the output neighborhood graphs for the real-world datasets we use in our experiments. Then, we examine the effectiveness of our method at choosing candidates and pruning the similarity search space. Finally, we compare the efficiency of our method against existing state-of-the-art exact and approximate baselines.

A. Neighborhood graph statistics

While sparsity of the input vectors plays a big role in the number of objects that must be compared to solve the APSS problem, the number of computed similarities is also highly dependent on the threshold ϵ. We studied properties of the output graph to understand how the input threshold can affect the efficiency of search algorithms. Each non-zero value in the adjacency matrix of the neighborhood graph represents a pair of objects whose similarity must be computed and cannot be pruned. A fairly dense neighborhood graph adjacency matrix means any APSS algorithm will take a long time to solve the problem, no matter how effectively it can prune the search space. Table II shows the average neighborhood size (μ) and neighborhood graph density (ρ) for six of the test datasets and ϵ ranging from 0.1 to 0.9. Graph density is defined here as the ratio between the number of edges (object pairs with similarity at least ϵ) and $n(n-1)$, which is the number of edges in a complete graph with n vertices.

TABLE II
NEIGHBORHOOD GRAPH STATISTICS

ϵ	μ	ρ	μ	ρ	μ	ρ
	WW500k		**WW100k**		**RCV1**	
0.1	1,749	3.5e-03	641	6.4e-03	10,986	1.4e-02
0.2	233	4.7e-04	101	1.0e-03	2,011	2.5e-03
0.3	64	1.3e-04	33	3.3e-04	821	1.0e-03
0.4	25	5.1e-05	16	1.7e-04	355	4.4e-04
0.5	10	2.2e-05	10	1.0e-04	146	1.8e-04
0.6	4.7	9.5e-06	6.3	6.3e-05	57	7.2e-05
0.7	2.1	4.2e-06	4.4	4.3e-05	25	3.2e-05
0.8	0.93	1.9e-06	2.9	2.9e-05	14	1.8e-05
0.9	0.28	5.7e-07	0.96	9.6e-06	8.1	1.0e-05
	Wiki		**Twitter**		**Orkut**	
0.1	801	4.4e-04	875	6.0e-03	76	2.5e-05
0.2	220	1.2e-04	259	1.8e-03	21	6.9e-06
0.3	74	4.1e-05	185	1.3e-03	7.2	2.4e-06
0.4	20	1.1e-05	138	9.5e-04	2.3	7.6e-07
0.5	7.6	4.2e-06	93	6.4e-04	0.69	2.3e-07
0.6	3.3	1.8e-06	49	3.4e-04	0.22	7.2e-08
0.7	1.7	9.6e-07	15	1.1e-04	0.09	3.1e-08
0.8	0.87	4.8e-07	2.8	1.9e-05	0.07	2.1e-08
0.9	0.35	1.9e-07	0.11	7.3e-07	0.06	2.0e-08

The table shows the average neighborhood size (μ) and neighborhood graph density (ρ) for the test datasets and ϵ ranging from 0.1 to 0.9.

As expected, the similarity graph is extremely sparse for high values of ϵ, with less than one neighbor on average in all but one of the datasets at $\epsilon = 0.9$. However, the average number of neighbors and graph density increase disproportionally for the different datasets as ϵ increases. The Orkut objects have less than 10 neighbors on average even at $\epsilon = 0.3$, while the RCV1 objects have more than 100 neighbors on average at $\epsilon = 0.5$. To put things in perspective, the **8.84 billion** edges of the RCV1 neighborhood graph for

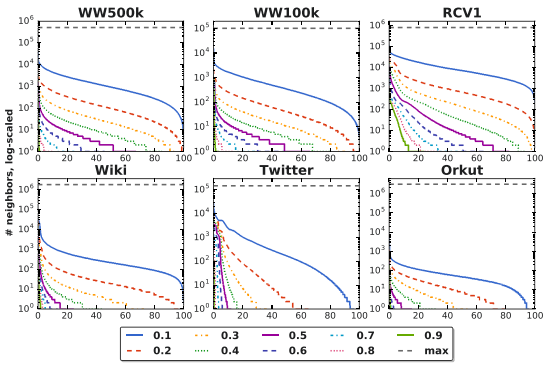


Fig. 1. Neighbor count distributions for several values of ϵ.

$\epsilon = 0.1$ take up **204 Gb** of hard drive space, and more than half of those represent similarities below 0.3, which are fairly distant neighbors. Nearest neighbor based classification or recommender systems methods often rely on a small number (often 1-10) of each object's nearest neighbors to complete their task. This analysis suggests that different ϵ thresholds may be appropriate for the analysis of different datasets. Searching the RCV1 dataset using $\epsilon = 0.8$, Twitter using $\epsilon = 0.7$, WW100 and WW500 using $\epsilon = 0.5$, Wiki using $\epsilon = 0.4$, and Orkut using $\epsilon = 0.2$ would provide enough nearest neighbors on average to complete the required tasks.

Figure 1 gives a more detailed picture of the distribution of neighborhood sizes for the similarity graphs in Table II. The *max* line shows the number of neighbors that would be present in the complete graph. The purple line for $\epsilon = 0.9$ is not visible in the figure for some datasets, due to the extreme sparsity of that graph. The vertical difference between each point on a distribution line and the *max* line represents the potential for savings in filtering methods, i.e., the number of objects that could be pruned without computing their similarity in full. Note that the y-axis is log-scaled. While there is a potential for pruning more than half of the objects in each object search in general, graph datasets show much higher pruning potential, especially given high minimum similarity thresholds.

B. Effectiveness of candidate choice and pruning

We instrumented our code to count the number of object pairs that were considered for similarity computation (# candidates) and the number of full executed dot-product/similarity computations (# dot-products). In Table III, we report the percent of candidates (*cand*) and dot-products (*dps*) executed by our method as opposed to those of a naïve APSS method ($n(n-1)/2$), for each of the six test datasets and ϵ ranging from 0.3 to 0.9. CANN was tuned to achieve 0.9 recall.

The results show that CANN is able to achieve high recall in computing the ϵNNG even while considering much fewer than 1% of the candidates in general. Moreover, most of the candidates are pruned by the L2-norm based filtering in CANN, resulting in 0.01% or fewer of the potential similarities being actually computed. The candidate selection and pruning effectiveness in our method lead to higher efficiency than competing methods, as we show in the next experiment.

TABLE III
CANDIDATE CHOICE AND PRUNING EFFECTIVENESS

ϵ	cand	dps	cand	dps	cand	dps
	WW100k		WW500k		RCV1	
0.3	0.2908	0.0380	0.1400	0.0152	0.4040	0.1058
0.4	0.1335	0.0176	0.0488	0.0060	0.2014	0.0521
0.5	0.0931	0.0094	0.0268	0.0022	0.1408	0.0271
0.6	0.0650	0.0045	0.0216	0.0010	0.1165	0.0134
0.7	0.1546	0.0057	0.0209	0.0004	0.0963	0.0058
0.8	0.3505	0.0042	0.0710	0.0002	0.1117	0.0040
0.9	0.3480	0.0012	0.1403	0.0001	0.0864	0.0019
	Twitter		Orkut		Wiki	
0.3	1.2240	0.2905	0.0063	0.0044	0.0194	0.0075
0.4	0.8944	0.1990	0.0045	0.0029	0.0100	0.0031
0.5	0.8007	0.1501	0.0029	0.0018	0.0087	0.0020
0.6	0.5810	0.0852	0.0018	0.0010	0.0055	0.0010
0.7	0.5374	0.0419	0.0009	0.0005	0.0042	0.0006
0.8	0.4131	0.0164	0.0003	0.0002	0.0025	0.0003
0.9	0.4736	0.0070	0.0003	0.0001	0.0020	0.0001

The table shows the percent of potential object comparisons (cand) and computed dot-products (dps) executed by our method as opposed to those of a naïve approach, when tuned to achieve 0.9 recall, for the test datasets and ϵ ranging from 0.3 to 0.9.

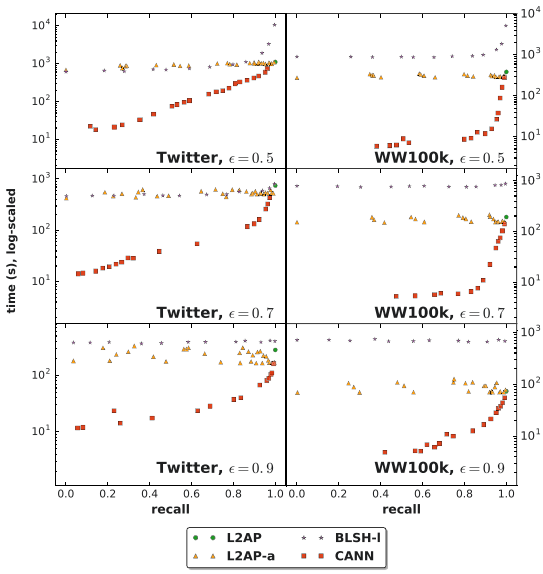


Fig. 2. Efficiency performance of the competing methods at different recall levels.

C. Execution efficiency

Figure 2 shows the efficiency of the competing methods at different recall levels, for two of the datasets and $\epsilon \in \{0.5, 0.7, 0.9\}$. L2AP is an exact method and is thus only present once in each sub-figure. The execution of both L2AP-a and BLSH-l are dominated by their respective candidate generation stage, and increasing the allowed error rate in the method seems to do little to improve the overall efficiency. In contrast, our method can be tuned, via the initial neighborhood size k and the candidate list size parameters μ_1 and μ_2, and can achieve over an order of magnitude performance improvement at lower recall levels as opposed to very high recall. CANN is still competitive at very high recall levels (e.g., 0.99) reaching execution times similar to the best exact method, L2AP.

The results of Data Mining algorithms are often not affected by an approximate nearest neighbor graph solution if average

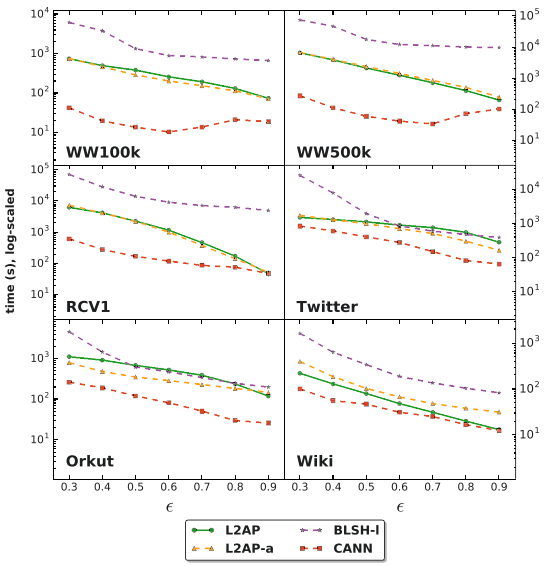


Fig. 3. Execution times for CANN and baselines for ϵ between 0.3 and 0.9, at minimum recall 0.9.

recall is sufficiently high [15], e.g., 0.9. As such, we compared the efficiency of all methods for this acceptable recall level and report results in Figure 3, for each of the six test datasets and ϵ between 0.3 and 0.9. As we also reported in [6], L2AP-a performs similarly to L2AP, and the exact method is sometimes faster than the approximate one due to time spent hashing in L2AP-a. The candidate generation in BLSH-l is not as competitive as the one in L2AP, and Bayesian pruning could not overcome the deficit in most cases, making BLSH-l the slowest baseline in general. CANN outperformed all baselines in all experiments except RCV1 at $\epsilon = 0.9$, where the speedup value was 0.97x. Speedup ranged between 0.97x–35.81x for text datasets, and 1.05x–6.18x for network datasets. Given the dataset-specific similarity thresholds suggested in Section V-A, which would result in at least 10 neighbors on average being included in the ϵNNG, CANN achieved an efficiency improvement of 2.1x–35.81x for the different datasets.

VI. Conclusion

In this paper, we presented CANN, an efficient approximate algorithm for constructing the cosine similarity graph for a set of objects. Our method leverages properties of the data and an initial incomplete neighborhood graph to prioritize choosing candidates for a query that are likely to be its neighbors. Furthermore, our method leverages recently developed filtering techniques to prune much of the search space both before and while computing candidate similarities. We conducted extensive experiments that measure both the effectiveness and efficiency of our method, compared to several state-of-the-art approximate and exact similarity graph construction baselines. Our method effectively reduces the number of candidates that should be compared to achieve a certain level or recall, and prunes many of the candidates without fully computing their similarity, resulting in up to 35.81x speedup over the best alternative method.

Acknowledgment

This work was in part made possible due to computing facilities provided by the Digital Technology Center (DTC) and the Minnesota Supercomputing Institute (MSI) at the University of Minnesota. We thank the reviewers for their helpful comments.

References

[1] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig, "Syntactic clustering of the web," in *Selected papers from the sixth international conference on World Wide Web*. Essex, UK: Elsevier Science Publishers Ltd., 1997, pp. 1157–1166.

[2] A. Metwally, D. Agrawal, and A. El Abbadi, "Detectives: Detecting coalition hit inflation attacks in advertising networks streams," in *Proceedings of the 16th International Conference on World Wide Web*, ser. WWW '07. New York, NY, USA: ACM, 2007, pp. 241–250.

[3] R. J. Bayardo, Y. Ma, and R. Srikant, "Scaling up all pairs similarity search," in *Proceedings of the 16th International Conference on World Wide Web*, ser. WWW '07. New York, NY, USA: ACM, 2007, pp. 131–140.

[4] G. Karypis, "Evaluation of item-based top-n recommendation algorithms," in *Proceedings of the Tenth International Conference on Information and Knowledge Management*, ser. CIKM '01. New York, NY, USA: ACM, 2001, pp. 247–254.

[5] S. Chaudhuri, V. Ganti, and R. Kaushik, "A primitive operator for similarity joins in data cleaning," in *Proceedings of the 22nd International Conference on Data Engineering*, ser. ICDE '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 5–.

[6] D. C. Anastasiu and G. Karypis, "L2ap: Fast cosine similarity search with prefix l-2 norm bounds," in *30th IEEE International Conference on Data Engineering*, ser. ICDE '14, 2014.

[7] V. Satuluri and S. Parthasarathy, "Bayesian locality sensitive hashing for fast similarity search," *Proc. VLDB Endow.*, vol. 5, no. 5, pp. 430–441, Jan. 2012.

[8] Y. Park, S. Park, S.-g. Lee, and W. Jung, "Greedy filtering: A scalable algorithm for k-nearest neighbor graph construction," in *Database Systems for Advanced Applications*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2014, vol. 8421, pp. 327–341.

[9] W. Dong, C. Moses, and K. Li, "Efficient k-nearest neighbor graph construction for generic similarity measures," in *Proceedings of the 20th International Conference on World Wide Web*, ser. WWW '11. New York, NY, USA: ACM, 2011, pp. 577–586.

[10] D. C. Anastasiu and G. Karypis, "L2knng: Fast exact k-nearest neighbor graph construction with l2-norm pruning," in *24th ACM International Conference on Information and Knowledge Management*, ser. CIKM '15, 2015.

[11] Y. Malkov, A. Ponomarenko, A. Logvinov, and V. Krylov, "Approximate nearest neighbor algorithm based on navigable small world graphs," *Information Systems*, vol. 45, pp. 61–68, 2014.

[12] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "Rcv1: A new benchmark collection for text categorization research," *J. Mach. Learn. Res.*, vol. 5, pp. 361–397, Dec. 2004.

[13] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?" in *WWW '10: Proceedings of the 19th international conference on World wide web*. New York, NY, USA: ACM, 2010, pp. 591–600.

[14] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *Proc. Internet Measurement Conf.*, 2007.

[15] J. Chen, H.-r. Fang, and Y. Saad, "Fast approximate knn graph construction for high dimensional data via recursive lanczos bisection," *J. Mach. Learn. Res.*, vol. 10, pp. 1989–2012, Dec. 2009.

Data Analytics through Sentiment Analysis

Information Extraction Engine for Sentiment-Topic Matching in Product Intelligence Applications

Cornelia Ferner, Werner Pomwenger, Stefan Wegenkittl
Salzburg University of Applied Sciences
Salzburg, Austria

Martin Schnöll, Veronika Haaf, Arnold Keller
Fact AI KG
Salzburg, Austria

***Abstract*—Product reviews are a valuable source of information for companies and customers alike. While companies use this information to improve their products, customers need it for decision support. Online shops often provide reviews, opinions and additional information to encourage customers to buy on their site. However, current online review implementations often lack a quick overview of how well certain product components meet customer preferences making product comparison difficult. Therefore, we have developed a product intelligence tool that combines state-of-the-art technologies into a natural language processing engine. The engine is capable of collecting and storing product-related online data, extracting metadata and analyzing sentiments. The engine is applied to technical online product reviews for component-level sentiment analysis. The fully automated process crawls the web for expert reviews, extracts sequences related to product components and aggregates the sentiment values from the reviews.**

Keywords—information extraction; topic detection; sentiment analysis; natural language processing; product intelligence;

I. INTRODUCTION

Making use of the web as a valuable source of opinions and assessments in product intelligence applications requires tools for the analysis of unstructured natural language data. Such text-based tools combine methods from natural language processing (NLP) with machine learning. However, e-commerce still lack out of the box tools or services for automatic extraction of information from business or product related web content. Our vision is to provide a product recommendation service that utilizes available online expert reviews to increase e-commerce sales. For our analysis, we focused on the consumer electronics sector, more specifically on laptops.

Customers face two challenges when using available product rating systems in online shops. These are the need to (1) evaluate specific product components and (2) to parse voluminous product reviews. Meeting the first challenge requires to evaluate specific component performance, rather than just overall product performance. For example, a laptop is assembled from several components that have varying influence on consumer purchasing. A customer may want a laptop with an adjustable keyboard-backlight and a very accurate trackpad, but be satisfied with an average sound quality. In order to evaluate the individual product components, it is necessary to identify any reference to those components in an article (so-called topic or aspect detection) and then to compute and match sentiments to these components.

The second challenge results from the overwhelming volume of detailed information found in online reviews. In common online shop applications, the user must read through customer reviews of varying quality and detail to find determine the accuracy of a laptop's touchpad. The impact of customer reviews have been extensively studied in word of mouth marketing [1], with paid and even fake reviews as negative side-effects [2]. Our product intelligence application (suitable for online shops) processes only high quality information taken from expert reviews from trusted product testing websites.

Online shops will benefit from this new component-based sentiment analysis as customers are provided with the desired product information directly on-site. Surveys[1] show that 63% of customers are more likely to purchase products from sites offering review information and that review content is more trusted than manufacturer descriptions. There is some evidence that customer reviews do not correlate with product quality [3] and hence support the idea of relying on expert information. At any rate, additional review-based product information leads to increased conversion and visitor return rates.

This paper introduces a fully automated Article and Review Information Extraction (ARIE) engine and its application in the context of online shops. Section III, describes ARIE's design and the subtasks performed for automated topic-wise sentiment matching. Section IV illustrates applications of the engine in the e-commerce domain. Section V shows that ARIE achieves state-of-the-art performance on each subtask and is comparable to human annotation, while automating this otherwise resource intensive task.

II. RELATED WORK

Topic detection, also known as aspect detection or topic analysis, is the task of identifying sequences in a text that refer to a certain topic. This can be done using both, supervised and unsupervised methods. The term topic detection should not be confused with document classification, where a whole document is assigned a topic, for example classifying a newspaper article as being a sports, financial or political article.

[1] Charlton, Graham. Ecommerce consumer reviews: why you need them and how to use them, 2015. https://econsultancy.com/blog/9366-ecommerce-consumer-reviews-why-you-need-them-and-how-to-use-them/

P. Haber et al. (Hrsg.), *Data Science – Analytics and Applications*, http://doi.org/10.1007/978-3-658-19287-7_7

Topic detection is used to determine the product components within the review text.

Sentiment analysis, also referred to as opinion mining, is the task of assessing the sentiment of a given text. This can either be done in a binary manner, evaluating the polarity of a statement and resulting in either a positive or a negative sentiment. More fine-grained sentiment analysis allows for multiclass classification, similar to a 5-star rating on review platforms.

Zhang and Liu [4] report on methods of aspect detection for opinion mining and Liu [5] gives a thorough overview of sentiment analysis. Schouten and Frasincar [6] provide an extensive overview of previous work on topic detection, sentiment analysis and aspect-based sentiment analysis in various domains.

Existing methods differ from our approach in various ways. References [7] and [8] describe unsupervised topic detection which is only useful if topics are not predefined. Others determine a single topic per document using topic models [9]. In 0, a single sentiment value per review is predicted by examining how sentences contribute to the overall rating rather than assigning a sentiment to each sentence. The polarity of words and phrases can be determined by rule-based approaches, such as using sentiment lexica [11], or syntax-based approaches, treating adjectives as opinion words [12]. In our engine, we learn the sentiments without using lexica. There is extensive work on sentiment analysis for movie reviews [13], [14]. The Stanford Sentiment Treebank[2] provides a comprehensive benchmark dataset. Other common sources for sentiment analysis are product reviews, including reviews about consumer electronics [7], 0, [11], [15]. Reference [16] focuses on analyzing social media for additional product intelligence information, but is limited to topic detection. Red Opal [17] is another addition to standard web shop solutions that generates product suggestions based on previous customer reviews.

Current solutions reach around 70% F1-score for both unsupervised and supervised topic detection tasks in reviews [7], [18]. State of the art for sentiment analysis on the Stanford movie review benchmark is reported to have 12.2% error rate for a binary sentiment classification and 51.3% for the 5-star classification, respectively [14].

This paper focuses on the application of topic-based sentiment analysis to a product intelligence application relying on high quality input in form of reviews from experts. Expert reviews mean longer text and more neutral language, thus various topics can be determined in a single review, where the expressed single sentiments are more difficult to extract.

III. Article and Information Extraction Engine

This section describes ARIE's design and the subtasks performed for automated topic-wise sentiment matching. Fig. 1 illustrates the information extraction process. Besides a crawler and a data storage, a framework for processing the collected websites is needed. As ARIE is embedded in a Java-based

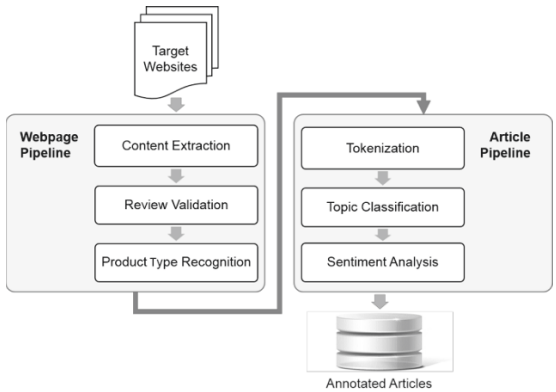


Fig. 1 ARIE. The engine crawls product-testing websites and processes the reviews in two pipelines sequentially. The webpage pipeline preprocesses the html files to extract valid review content and recognizes the product type. The article pipeline works on each review to classify topics and sentiments.

environment, we decided to employ the Unstructured Information Management Architecture (UIMA) framework [19] for the natural language processing and annotation tasks. UIMA is a Java framework originally developed by IBM and now maintained by the Apache Software Foundation[3]. Consistently, Apache Nutch2[4] can be used as crawler and Apache Cassandra[5] as storage solution. In the data acquisition phase, the crawler was targeted at selected online testing platforms[6] resulting in a current total of over 3600 expert product reviews that were then available for further analysis.

The text analysis components are called "annotators" in UIMA. The openNLP[7] library is included in UIMA, offering out-of-the-box annotators for the most common NLP tasks such as tokenization or sentence segmentation. By extending a base class, custom annotators can be integrated in UIMA. Several annotators can be combined into an annotation pipeline. In ARIE, two pipelines are implemented: A low-level pipeline filters relevant websites and extracts metadata and the raw article text (see Fig. 1, "Webpage Pipeline"). A high-level pipeline brakes up the article contents with respect to product components and analyzes and classifies those parts (see Fig. 1, "Article Pipeline"). Reviews are stored in an XML-like structure and all annotations are added to the base file with additional tags.

A. The Webpage Pipeline

The first step in the Webpage pipeline is the parsing of the webpages and the extraction of relevant content using html tags. An important part of this task is to remove boilerplate information - content in the header, footer or navigation bar of the web page. ARIE uses boilerpipe[8], a recent Java library

[2] Stanford Sentiment Treebank available at: http://nlp.stanford.edu/sentiment

[3] Apache UIMA https://uima.apache.org/
[4] Apache Nutch available at: https://nutch.apache.org/
[5] Apache Cassandra available at: http://cassandra.apache.org/
[6] Such as e.g. provided by http://pcworld.com/
[7] OpenNLP available at: https://opennlp.apache.org/
[8] Boilerpipe API available at: https://boilerpipe-web.appspot.com/

providing algorithms for isolating a website's main text, as content extraction annotator.

Classifier stacking is then used to implement an annotator for validating the reviews and rejecting non-product reviews such as summaries of technical details or news articles. Stacking classifiers is an ensemble learning method, which evolved out of the concept of stacked generalization [20] for neural networks. The idea is to train a classifier on a single task (review validation) and have a super classifier [21] make the final decision by using the review URL and specific keywords as additional features. Latent dirichlet allocation (LDA) [22], a topic model for discovering semantic structure in the reviews, is used as the base classifier. The binary super classifier is a support vector machine (SVM) [23].

Product type recognition is the last subtask of the Webpage pipeline. It assigns the review to one of the categories laptop, tablet and other products. The annotator is a document-level maximum entropy (MaxEnt) [24] classifier, a variant of logistic regression for multiclass problems.

B. *The Article Pipeline*

The first step necessary in analyzing the article content is the tokenization of the raw text. UIMA provides annotators for tokenizing text into sentences and words. The sentence-level tokenization relies on punctuation characters, while the word-level tokenizer treats whitespace as word boundary. Tokenization can also be applied on paragraph-level by detecting line breaks using regular expressions. Tokenization on several levels allows for the application of different classification methods and for more or less fine-grained results.

Topic classification on sentence level is used to assign a topic (that is, a product component) to each sentence. We use a Hidden Markov Model (HMM) [25] as the topic detection algorithm. HMMs require transition and emission probabilities as well as the initial distribution as input features. The transition of the topics (in which sequence the topics appear) and the distribution of initial topics in the reviews are estimated from the corpus. For the emission probabilities (the probability of a word appearing in a specific topic), we implemented a novel estimator based on a MaxEnt classifier [26] – At first, a MaxEnt classifier is trained on the predefined product components. The internal topic weights of the trained MaxEnt classifier are then converted into emission probabilities for the HMM that is again applied on the corpus.

TABLE I. LAPTOP COMPONENTS (TOPICS)

Topic	Relative Frequency	Topic	Relative Frequency
Build/Case	10.78%	Specifications	8.15%
Display	10.00%	Introduction	3.88%
Sound	2.94%	Offtopic	1.51%
Keyboard	4.97%	Review Info	0.37%
Touchpad	3.52%	Software	3.29%
Noise	2.10%	Summary/Verdict	14.92%
Temperature	2.61%	Warranty	0.95%
HW/Performance	23.87%	Webcam	0.72%
Battery/Power	5.42%		

By using an HMM, the inherent structure of review articles can be exploited: Some topics are more likely to appear at the beginning of the review. Besides the *introduction*, very frequently the laptop *case* is discussed first. *Summary/verdict* or the *review info* are topics that often appear at the end of a review. Some topics, such as *keyboard* and *touchpad*, are likely to follow each other. As the HMM operates on word-level, topic changes are infrequent, resulting in a diagonal dominated transition matrix. Furthermore, the HMM can be used to assign topics at sub-sentence level which is useful in comparative sentences: The sentence "The laptop is very fast but also noisy." would ideally be assigned two topics, then: *performance* and *noise*. This detailed, domain-specific information cannot be incorporated into a stand-alone MaxEnt classifier.

Different types of deep neural networks, such as recurrent (RNN) and recursive neural tensor networks (RNTN) were evaluated for sentiment analysis. RNTNs come at the cost of requiring a parsed syntax tree as input and are hierarchical networks. RNTNs yield good performance assigning sentiments on word-level [27]. Performance evaluation on sentence-level suggests using long short-term memory [28] (the deep learning variant of a recurrent neural network) to predict the sentiments. RNNs directly work on the tokenized sentence as sequential input. The classified sentiments are aggregated over sentences, in order to assign them to a product component.

In summary, the article pipeline outputs component-sentiment pairs that can be aggregated based on the product type for all validated product reviews. The results from ARIE can be used to generate summaries for each topic or to give a visual overview of the "performance" of each product component.

IV. PRODUCT INTELLIGENCE WITH ARIE

In the following section, detailed information on how ARIE is trained on laptop reviews for extracting component-sentiment matches is given. ARIE is currently set up for reviews in English that have been manually labeled by the fact.ai company. For all subtasks of the two pipelines, specific datasets were sampled out of the crawled websites.

The review validation task was performed on a random sample of 3605 product reviews and 2686 non-product reviews (with technical content), which were available for training and testing. The sample set for the task product type recognition comprises 499 tablet reviews, 501 laptop reviews and 393 reviews about other technical products.

TABLE II. SENTIMENT CLASSES

Sentiment Class	Relative Frequency	Aggregation Weight
Very negative	4.56%	1.0
Negative	28.13%	0.6
Neutral	16.89%	0.2
Positive	39.54%	0.6
Very positive	10.89%	1.0

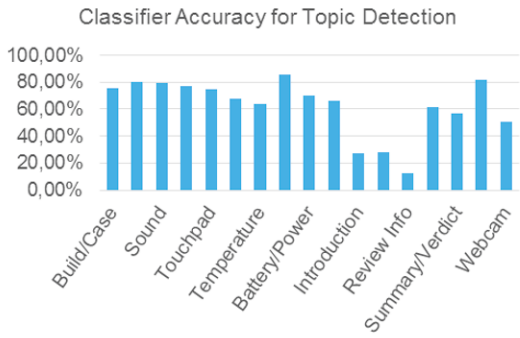


Fig. 2 Topic detection accuracy of the stand-alone MaxEnt classifier on all seventeen topics.

The corpus for topic detection comprised 3152 reviews, of which 240220 sentences were labelled manually. 17 topics were pre-defined, not all of them related directly to the laptop. TABLE I lists all topics and their relative frequency in the corpus. It is obvious that the topic distribution is non-uniform. Some topics, such as specifications, are regularly discussed in laptop reviews as they are vital for the rating. Others, such as webcam, only occur if the laptop has this feature. Nonetheless, these are interesting aspects to analyze.

The corpus for sentiment analysis consisted of 21695 manually labelled sentences extracted from laptop reviews, both on sentence and word level. Two forms of analysis were applied: a fine-grained 5-class and a binary analysis. The 5-class rating includes the classes very positive, positive, neutral, negative and very negative. For the binary classification, the neutral class was omitted and the two positive and the two negative classes were combined.

For each sentence, the assigned topic was then matched with the corresponding 5-class sentiment and aggregated to get an average score for each product review (vertical aggregation). Additionally, sentiment values were aggregated over all reviews to get a component-based average value for a specific product (horizontal aggregation). As many sentences in review texts are neutral and do not contribute to the overall sentiment (see TABLE II), a weighting scheme was applied, where the positive, negative and neutral classes are penalized.

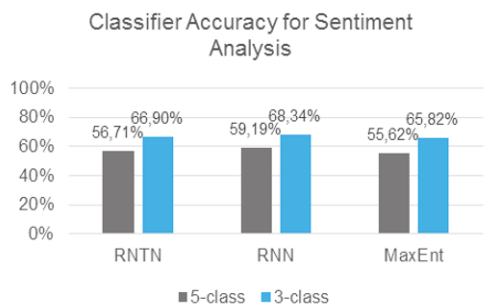


Fig. 3 Sentiment accuracy for the 5-class (left) and 3-class (right) setting.

V. Evaluation

The performance of ARIE is reported in the aforementioned settings, with a more detailed focus on topic classification and sentiment analysis, as those are the essential tasks in component-sentiment matching.

We achieved an accuracy of 99% in review validation, using the stacked classifier setting. The product type recognition achieved an accuracy of 96.97%, where most misclassifications were due to reviews describing convertible devices (hybrids out of laptop and tablet). A definite classification as laptop or tablet was infeasible for those reviews.

The MaxEnt topic classification on sentence level yields an overall accuracy of 71.04% and an average F1-score of 66.35% on the laptop test set. A closer look on the specific topic results (see Fig. 2) reveals that topics related to the review itself (*introduction*, *offtopic*, *review info* and *summary*) perform much worse than laptop related topics. The average F1-score is not higher than 36.61% for those four topics. Combining the four non-technical topics to a single class for the performance evaluation increases the overall accuracy to 73.39% and the F1-score to 75.59%.

The word weights learnt by the MaxEnt algorithm were then incorporated as emission probabilities for the HMM, which boosted the overall accuracy by 3.9%. The HMM F1-score for only the laptop related topics increased to 78.9%. As becomes obvious from their individual results, the performance increase is even higher for the technical topics (see Fig. 4).

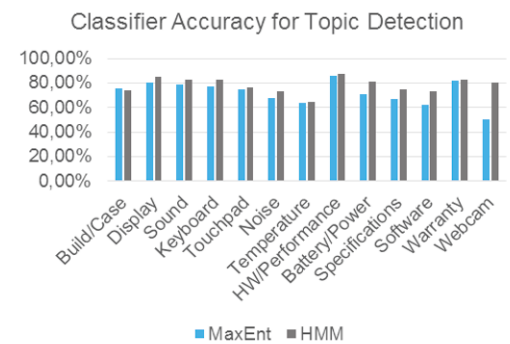


Fig. 4 Comparison of the topic detection accuracy of the stand-alone MaxEnt (left) and the HMM (right) for technical topics only.

Performance evaluation for sentiment analysis was also reported on sentence level rather than on the aggregated overall values. Difficulties distinguishing between the two negative and the two positive classes were also observed with human annotators and were evident in the classification results. Therefore, we also evaluated the performance of the classifiers on a ternary basis, where the two positive and the two negative classes were combined. The RNN outperformed both a standard MaxEnt classifier and the RNTN (see Fig. 3) yielding 68.34% for the ternary setting.

Evidence data is available from a test web shop that implemented our technology and directly visualized the aggregated sentiment values for laptops, during a three month trial period. Note that the review information was only integrated for a fraction of laptops provided in the web shop. According to the figures in TABLE III, customers visiting a laptop site that had details on topics and sentiments available were more likely to purchase the product. This relates to an increase of about 44% in conversion rate. However, we are aware that other factors can influence the conversion rate as well and would require a longer trial period to confirm these numbers.

TABLE III. TEST WEB SHOP CONVERSION RATE

	Number of laptops viewed	Number of then purchased laptops	Conversion rate
Without review information	1828	243	13.29%
With review information	422	81	19.19%

VI. CONCLUSION

We demonstrated the application of NLP methods to extract the inherent information in online expert reviews by combining and modifying algorithms for the evaluation of review content: review validation, product type recognition, topic detection and sentiment analysis. We identified a lack of readily available tools for building our engine ARIE, so we combined a set of handcrafted methods suitable to solve the problems at hand. This proved that an automated processing of the tasks is possible and yields state-of-the-art performance.

With some additional effort, the results presented from sentiment-topic matching can be visualized in online shops and serve as customer guidance. Furthermore, the extracted information offers valuable insight for companies. Businesses can compare products to those of competitors and identify weak points to improve during product development. There is still work to be done concerning the generalization of the process for other product and product categories.

ACKNOWLEDGMENT

This work was funded by the Austrian Research Promotion Agency (FFG) in the FME project (FFG No. 853848).

REFERENCES

[1] J. Chevalier and D. Mayzlin, "The effect of word of mouth on sales: Online book reviews," Journal of Marketing Research, vol. 43(3), pp. 345-354, 2006.

[2] D. Mayzlin, Y. Dover and J. Chevalier, "Promotional reviews: an empirical investigation of online review manipulation," The American Economic Review, vol. 104(8), pp.2421-2455, 2014.

[3] B. de Langhe, P. Fernbach and D. Lichtenstein, "Navigating by the stars: investigating the actual and perceived validity of online user ratings," Journal of Consumer Research, vol. 42(6), pp.817-833, 2016.

[4] L. Zhang and B. Liu, "Aspect and entity extraction for opinion mining," Data mining and knowledge discovery for big data, pp. 1-40, 2014.

[5] B. Liu, "Sentiment analysis and opinion mining," Synthesis lectures on human language technologies, vol. 5(1), pp. 1-167, 2012.

[6] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," IEEE Transactions on Knowledge and Data Engineering, vol. 28(3), pp. 813-830, 2016.

[7] J. Yu, Z. Zha, M. Wang and T. Chua, "Aspect ranking: identifying important product aspects from online consumer reviews," Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, ACL, 2011.

[8] A. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," Natural Language Processing and Text Mining, Springer London, pp. 9-28, 2007.

[9] N. Srivastava, R. Salakhutdinov and G. Hinton, "Modeling documents with a deep boltzmann machine," Uncertainty in Artificial Intelligence, 2013.

[10] N. Pappas and A. Popescu-Belis, "Explaining the stars: weighted multiple-instance learning for aspect-based sentiment analysis," Conference on Empirical Methods in Natural Language Processing, 2014.

[11] C. Zirn, M. Niepert, H. Stuckenschmidth and M. Strube, „Fine-grained sentiment analysis with structural features," Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP), pp. 336-344, ACL, 2011.

[12] M. Hu and B. Liu, "Mining and summarizing customer reviews," Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discover and Data Mining, ACM, pp. 168-177, 2004.

[13] B. Pang and L. Lee, "Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales," Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, ACL, 2005.

[14] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," Proceedings of ICML, vol. 14, 2014.

[15] X. Fang and J. Zhan, "Sentiment analysis using product review data," Journal of Big Data, vol. 2(5), 2015.

[16] A. Abrahams, J. Jiao, W. Fan, G. Wang and Z. Zhang, "What's buzzing in the blizzard of buzz? Automotive component isolation in social media postings," Decision Support Systems, vol. 55(4), pp. 871-882, 2013.

[17] C. Scadiffi, K. Bierhoff, E. Chang, M. Felker, H. Ng and Ch. Jin, "Red Opal: Product-feature scoring from reviews," Proceedings of the 8th ACM Conference on Electronic Commerce, ACM, pp. 182-191, 2007.

[18] N. Jakob and I. Gurevych, "Extracting opinion targets in a single- and cross-domain setting with conditional random fields," Conference on Empirical Methods in Natural Language Processing, 2010.

[19] D. Ferrucci and A. Lally, "UIMA: An architectural approach to unstructured information processing in the corporate research environment," Natural Language Engineering, 10(3), pp. 237-348, 2004.

[20] D. Wolpert, "Stacked generalization," Neural Networks, vol. 5(2), pp.241-259, 1992.

[21] M. Van der Laan, E. Polley and A. Hubbard, "Super learner," Statistical applications in genetics and molecular biology, vol. 6(1), 2007.

[22] D. Blei, A. Ng and M. Jordan, "Latent dirichlet allocation," Journal of Machine Learning Research, pp.993-1022, 2003.

[23] B. Boser, I. Guyon and V. Vapnik, "A training algorithm for optimal margin classifiers," Proceedings of the 5th Annual Workshop on Computational Learning Theory, ACM, 1992.

[24] A. Berger, S. Della Pietra and V. Della Pietra, "A maximum entropy approach to natural language processing," Computational Linguistics, vol. 22(1), pp. 39-71, 1996.

[25] Cappé, O., Moulines, E., and Ryden, T., "Inference in Hidden Markov Models," New York, Springer, 2005.

[26] C. Ferner, and S. Wegenkittl, "Maximum Entropy Based Emission Probabilities in Higher Order Hidden Markov Models," unpublished.

[27] Socher, R., Perelygin, A., Wu. J., Chuang, J., Manning, C., Ng. A, and Potts, C., "Recursive deep models for semantic compositionality over a sentiment treebank," Conference on Empirical Methods in Natural Language Processing, 2013.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9(8), pp.1735-1780, 1997.

Towards German Word Embeddings: A Use Case with Predictive Sentiment Analysis

Christian Bauckhage, Eduardo Brito,
Konstadin Cvejoski, César Ojeda and Rafet Sifa
Fraunhofer IAIS
St. Augustin, Germany

Rafet Sifa and Christian Bauckhage
University of Bonn,
Bonn, Germany

***Abstract*—Despite the research boom on words embeddings and their text mining applications from the last years, the vast majority of publications focus only on the English language. Furthermore, hyperparameter tuning is a rarely well documented process (specially for non English text) that is necessary to obtain high quality word representations. In this work, we present how different hyperparameter combinations impact the resulting German word vectors and how these word representations can be part of more complex models. In particular, we perform first an intrinsic evaluation of our German word embeddings, which are later used within a predictive sentiment analysis model. The latter does not only serve as an extrinsic evaluation of the German word embeddings but also shows the feasibility of predicting preferences only from document embeddings.**

I. Introduction

Representing words as vectors of real numbers has become a common technique to solve various Natural Language Processing (NLP) tasks including machine translation [1], semantic and syntactic language modeling [2]; and named entity recognition [3]. Despite the increasing number of publications about these distributed word representations and their applications, most of the literature focuses on learning word embeddings for texts in English. Moreover, although some authors provide recommendations about reasonable hyperparameter ranges for their respective models, it is not clear whether they hold for non English corpora.

In the first part of our work, we train German embeddings from SdeWaC, a large German corpus created by web-crawling the ".de" domain [4], [5]. We train 11 word vector models with different hyperparameter combinations and we evaluate them on a similarity task in order to check whether the general recommendations given for English word embeddings apply for German as well. Taking the generated word embeddings from the previous task as starting point, we map Google Play reviews to the same vector space using the Paragraph Vector model [6]. We use these new vectors to predict whether a user liked an app given a review with three different algorithms: logistic regression, decision trees and random forests. Although there is no correlation between our quality measure at a word representation level (the results of the word similarity evaluation) and at a prediction level (geometric mean of accuracy), we checked that our best word embedding models can lead to acceptable prediction models.

II. Contribution

Our work aims to clarify the impact of the hyperparameters on the quality of word embeddings, which is in general a poorly documented technical detail, specially for non English embeddings. Additionally, we evaluate paragraph embeddings that are inferred from word representations learned from the previous step. The evaluation is casted as a supervised learning problem aiming to predict labels assigned to text based on the learned representations. Moreover, this shows the potential of this approach allowing to analyze inherently rich text corpora for Business Intelligence. Using this approach, stakeholders at any company can build predictive models over the text base that is augmented by social media content to have more insights about their customers. Namely, combined with the methods from the mature field of predictive analytics, having numerical data representations for chunks of customer text will allow us to analyze trends, implicit and explicit user feedback and future interest in company products. For that, we present a case study to predict implicit user feedback of German Google Play reviews. As a whole, this work must be understood as an initial feasibility study about performing sentiment analysis on informal and noisy German texts by means of word representations. Further optimization of our best models is left for future work.

III. Related Work

Assuming that words that occur in similar contexts tend to have similar meanings (the so-called *distributional hypothesis* [7]), several models exploiting word co-occurrence have been developed to find word representations. Among them, *word embeddings* are distributed vector representations, which are dense, low-dimensional, real-valued and can capture latent features of the word [8]. These word vectors encode syntactical and semantical information so that some NLP tasks can be solved by simple linear vector operations thanks to the distributed nature of the word representations. For example, we can answer analogy questions just with additions and subtractions: if we subtract from a vector representing the word "Madrid" the vector corresponding to "Spain" and we add the vector "France", the resulting vector should be very close to the vector "Paris" [2]. These vector word representations have also the advantage that they can be trained efficiently by simple (shallow) neural networks such as the continuous Skip-gram model (SG) and the Continuous Bag of Words model (CBOW)

P. Haber et al. (Hrsg.), *Data Science – Analytics and Applications*, http://doi.org/10.1007/978-3-658-19287-7_8

[9], both popularized after the release of the `word2vec`[1] package.

The SG network is formally defined to predict nearby words. However, we focus on the the resulting hidden layer weights after the training phase: they constitute dense vector representations of words. Words are presented to the network with one-hot encoding. We can model a projection from the one-hot vector to its embedding by means of an identity transfer function. In contrast to the most common neural networks models which require a non-linear transfer function in the hidden layer, the SG model does not use any non-linear transfer function in the hidden layer but only in the output layer. The number of necessary neurons in the hidden layer is determined by the number of features that we want our word embeddings to have: if V is size of the vocabulary and N the size of our word vectors, we can represent the weights as a $V \times N$ matrix, in which each row i is the embedding of the corresponding word placed in the position i within the vocabulary. The context of a word token is defined by setting a maximum window size m so that for a token sequence w and a token in position t, the context of the token $w(t)$ is made of tokens that are at a maximum distance m' from the central word $w(t)$ (excluding the central word itself), where m' is sampled from the interval $[1, m]$. By choosing the window size stochastically for each example, we ensure that word tokens that appear closer to the central word get more importance, as they are more likely to fit within the context window

$$w(t - m') \cdots w(t - 1)w(t + 1) \cdots w(t + m'). \quad (1)$$

The output layer performs multinomial logistic regression (without bias term). Since we do not care about the prediction accuracy but about the quality of the word embeddings, the original formulation using softmax as output layer transfer function is simplified by using *negative sampling* or *hierarchical softmax* [2]. This is specially important to efficiently train the embeddings since all matrix calculations needed for the softmax activation function are computationally expensive.

The CBOW model works in a similar way but defined for the inverse task: in this case, the context words are the input of the network, which are used to predict the central word they surround. Also this prediction task is only used to learn the hidden layer weights, which correspond to word embeddings exactly as in the SG model.

The *Paragraph Vector* model [6] extends both mentioned models so that also a set of words such as a sentence or a whole document can be represented as a vector (the so-called paragraph vector). Paragraph vectors can be derived using two different architectures: the Distributed Bag of Words (PV-DBOW) and Distributed Memory (PV-DM) of Paragraph Vectors, which are respectively similar to SG and CBOW. They consider each paragraph as another token belonging to the vocabulary that appears in all context windows during the training phase. For more details about these models we refer the reader to [6].

[1]https://code.google.com/archive/p/word2vec

TABLE I
WORD2VEC HYPERPARAMETERS

Hyperparameter	Meaning
`cbow`	Network architecture: 0 for SG, 1 for CBOW
`window`	Maximum skip length between words within a context window (maximum window size)
`sample`	Threshold for word subsampling
`hs`	Hierarchical Softmax
`negative`	Number of negative samples
`min-count`	Word types below this count value are discarded

IV. Data

We use two different datasets to infer embeddings: SdeWaC [4] for training word embeddings for the German language and SCARE [10] to produce document embeddings for sentiment analysis. For evaluating the representation quality of the word embeddings, we use the Gur350 dataset [11], [12].

A. SdeWaC

The SdeWaC corpus is a subset of the deWaC corpus, a web-crawled collection of German texts extracted from the ".de" domain in the scope of the WaCky project [5]. It consists of 846.159.403 word tokens in 44.084.442 sentences, including 1.094.902 different word types. Thanks to its variety and size, this corpus is suitable to generate general purpose embeddings to model the German language. We apply a minimal pre-processing to this corpus before it is processed: it is only tokenized, lowercased and shuffled, taking a sentence as a text unit.

B. SCARE

The Sentiment Corpus of App Reviews (SCARE) consists of 802,860 German app reviews collected from Google Play Store. These reviews are divided in 11 different app categories: instant messengers, fitness trackers, social network platforms, games, news applications, alarm clocks, navigation and map applications, office tools, weather apps, sport news and music players. From each category, between 10 and 15 different apps are considered. All these reviews contain a text and a rating with a 1-5 star score. Both are used for our sentiment analysis task.

C. Gur350

The Gur350 dataset contains 350 pairs of German words with a human-annotated semantic relatedness score which was originally introduced to measure distances between words [11], [12].

V. Experiments

A. Word Embedding Benchmarking

We make use of the `word2vec` package [2] to obtain vector representations of German words. It learns word embeddings with mini-batch asynchronous stochastic gradient descent on a shallow neural network, which can have two possible architectures: continuous Skip-gram (SG) architecture or continuous bag-of-words (CBOW) [9]. With the purpose to check the effect of SG and CBOW hyperparameters that need

TABLE II
HYPERPARAMETERS USED FOR THE 11 TRAINED WORD VECTOR MODELS

Model	cbow	window	sample	hs	negative	min-count
0	0	8	0	1	10	50
1	1	8	0	1	10	50
2	0	5	0	1	10	50
3	0	15	0	1	10	50
4	0	8	1e-5	1	10	50
5	0	8	1e-3	1	10	50
6	0	8	0	0	10	50
7	0	8	0	1	0	50
8	0	8	0	1	20	50
9	0	8	0	1	10	10
10	0	8	0	1	10	100

TABLE III
SPEARMAN'S ρ BETWEEN THE TRAINED WORD EMBEDDINGS AND GUR350 WORD PAIRS HUMAN-ANNOTATED SEMANTIC RELATEDNESS

Model	ρ
0	0.7153
1	0.5510
2	0.6933
3	0.7325
4	0.7479
5	0.7335
6	0.7399
7	0.7002
8	0.7103
9	0.7222
10	0.7249

to be specified in `word2vec`, we train 11 different models on the SdeWaC corpus. A description of the hyperparameters is presented in Tbl. I and the tested values for them are shown in Tbl. II.

All models were set to generate vectors of size 300, which is a frequent value among publications about word embeddings. Since the usually recommended ranges for the hyperpameters may not apply for the German language (most of the available publications focus on the English language), all models from 1 to 10 differ only in one hyperparameter compared to model 0 so that the effect of each hyperparameter can be assessed[2].

The resulting embeddings are evaluated on a similarity task. The cosine distance is our similarity measure between our word vectors. By calculating the Spearman's rank correlation coefficient [13] between the cosine distances of word vector pairs and the semantic relatedness from the Gur350 dataset, we can evaluate the quality of our word embeddings.

From Tbl. III we can observe that the model with best performance (model 4) on the similarity evaluation corresponds to the one subsampling the most frequent words with highest threshold (1e-3). Considering that the other model applying subsampling (1e-5) obtained the third best Pearson's correlation, the recommendation of introducing subsampling during the training phase seems to be also valid for German embeddings. Besides that, we see that the only model applying the CBOW model (model 1) is clearly worse than the rest.

[2]Model 0 corresponds to the only previous work that we could find about German word embeddings in which hyperparameter settings are explicitly specified: https://github.com/devmount/GermanWordEmbeddings

TABLE IV
GEOMETRIC MEAN OF NEGATIVE AND POSITIVE CLASS ACCURACY VALUES OF CLASSIFYING OF LIKING OF FITNESS TRACKER APPLICATIONS

Setting	LR	DT	RF1	RF2
S0	0.791	0.677	0.729	0.696
S1	0.766	0.650	0.661	0.645
S2	0.789	0.685	0.730	0.695
S3	0.788	0.679	0.725	0.686
S4	0.784	0.689	0.729	0.691
S5	0.791	0.674	0.727	0.697
S6	0.788	0.685	0.719	0.693
S7	0.790	0.684	0.734	0.699
S8	0.783	0.680	0.726	0.693
S9	0.799	0.679	0.732	0.693
S10	0.789	0.668	0.724	0.693

Although one observation does not suffice to claim that SG outperforms CBOW, our result is in line with previous observations that the CBOW model cannot produce better word embeddings than SG in spite of being a more expressive model [14]. Regarding the window size, we confirmed our expectations that larger window sizes lead to better results. Therefore, model 3 (window size 15) improves model 0 (window size 8) whereas model 2 (window size 5) drops the Spearman's correlation.

B. Predictive Sentiment Analysis

We evaluate our word representations with a Business Intelligence use case which involves predicting user preferences from a given text document. This does not only help us to gain insight about the hyperparameter space for the learning process of word embeddings, but also shows that we can obtain high accuracy values for predicting user decisions by only using paragraph embeddings as data input. We train document representations for German Google Play Reviews from [10] for the fitness trackers category. The dataset contains 22188 anonymized reviews with ranking ranging from one to five. We binarized (or implicitized) the review score by assigning *False* class to ratings one, two and three and *True* class to ratings four and five to respectively model the notion of user's not liking and liking of the product. After this pre-processing step the resulting dataset contained a 26/74 class distribution ratio.

Similar to [6], we learn document representations of the reviews by means of the *Paragraph Vector with Distributed Bag of Words* model (PV-DBOW) [6] implemented in gensim `doc2vec`[3]. Since PV-DBOW does not explicitly learn word embeddings but only the paragraph vectors (document embeddings), the `doc2vec` implementation initializes the word vectors randomly if not specified otherwise. Although the learning algorithm should be able to work with this setting, the performance degrades severely in practice [15]. Therefore, we initialize our model with the pre-trained word vectors trained with the large external corpus SdeWaC that we obtained from our experiment explained in Sec. V-A. Then, we train 300-dimensional document vectors during 1000 epochs using PV-DBOW with a maximum window size of 15 and 5 negative

[3]https://github.com/RaRe-Technologies/gensim

samples per context window. We also subsample the most frequent words setting the threshold to 1e-5. By keeping this hyperparameter combination fixed, we can assess which of the pre-trained word embedding models lead to a better performance for our predictive sentiment analysis task.

For the supervised learning task of sentiment analysis, we used Logistic Regression (LR), Decision Trees (DT) and Random Forests with 101 and 11 random trees (RF1 and RF2 respectively). In Tbl. IV we show the evaluation of the predictions using 10-fold cross-validation in terms of the geometric mean of the accuracy values of both of the classes.

Overall, Logistic Regression yielded the best results for predicting the user's preferences with respect to the geometric mean accuracy. When we compare the experimental settings we observe that reducing the minimum count of words to be considered in the learning phase improves the representation with larger window sizes reaching almost up to 0.8 percent of the geometric mean for positive and negative classes.

It is also noticeable that the CBOW word embedding model has not performed well in terms of representation learning compared to all the SG models, which as already indicated by [14] for the English language among. However, the results on the sentiment analysis correlate poorly with the intrinsic evaluation using the Gur350 dataset. This fact is in line with very recent research on evaluating word embeddings showing that most word similarity datasets available for word similarity evaluations are not useful to predict a good performance on an extrinsic task such as sentiment analysis [16]. Our results suggest that Gur350 also suffers from this problem and it is thus not suitable to be used as an intrinsic evaluation dataset for German word embeddings.

VI. Conclusion and Future Work

We observed that some of the general recommendations to learn word embeddings in English (such as those from [14]) also apply for the German language when they are evaluated on a word similarity task: the continuous skip-gram model outperforms the continuous bag of word models and negative sampling provides better results than hierarchical softmax, where more negative samples improve the result. Furthermore, we detected that the threshold to downsample the most frequent words has the highest impact on the similarity score among all tested hyperparameters. Nonetheless, we also noticed that the models that perform the best on the similarity task do not necessarily provide the best contributions when they are used to infer document embeddings for the predictive sentiment analysis task. Like [16] does for the English language, we suggest not to rely on similarity tasks to assess the quality of German word embeddings until a suitable evaluation dataset is available. We also showed the feasibility of performing sentiment analysis on informal German text using document embeddings as features of the classifiers.

As a future work, we will focus on optimizing the sentiment analysis approach and the document representations for Business Intelligence applications in the areas of Game Analytics [17] and Web Intelligence [18]. It would be also interesting to check if our findings generalize across other German text collections and with other text pre-processing approaches.

Acknowledgement

The authors would like to express their gratitude to M. Sänger, U. Leser, S. Kemmerer, P. Adolphs, and R. Klinger for sharing the German Google Play Dataset; and M. Baroni, S. Bernardini, A. Ferraresi, and E. Zanchetta for making their web-crawled corpora available.

References

[1] W. Y. Zou, R. Socher, D. M. Cer, and C. D. Manning, "Bilingual word embeddings for phrase-based machine translation." in *EMNLP*, 2013, pp. 1393–1398.

[2] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[3] A. Passos, V. Kumar, and A. McCallum, "Lexicon infused phrase embeddings for named entity resolution," in *Conference on Computational Natural Language Learning (CoNLL)*, 2014.

[4] G. Faaß and K. Eckart, "SdeWaC–a corpus of parsable sentences from the web," in *Language processing and knowledge in the Web*. Springer, 2013, pp. 61–68.

[5] M. Baroni, S. Bernardini, A. Ferraresi, and E. Zanchetta, "The WaCky Wide Web: A collection of very large linguistically processed web-crawled corpora," *Language Resources and Evaluation*, vol. 43, no. 3, pp. 209–226, September 2009.

[6] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents." in *ICML*, vol. 14, 2014, pp. 1188–1196.

[7] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[8] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: A simple and general method for semi-supervised learning," in *48th Annual Meeting of the Association for Computational Linguistics*, A. for Computational Linguistics, Ed., 2010, pp. 384–394.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[10] M. Sänger, U. Leser, S. Kemmerer, P. Adolphs, and R. Klinger, "SCARE – The Sentiment Corpus of App Reviews with Fine-grained Annotations in German," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC16)*. Portorož, Slovenia: European Language Resources Association (ELRA), May 2016.

[11] I. Gurevych, "Using the structure of a conceptual network in computing semantic relatedness," in *International Conference on Natural Language Processing*. Springer, 2005, pp. 767–778.

[12] T. Zesch and I. Gurevych, "Automatically creating datasets for measures of semantic relatedness," in *Proceedings of the Workshop on Linguistic Distances*. Association for Computational Linguistics, 2006, pp. 16–24.

[13] C. Spearman, "The proof and measurement of association between two things," *The American journal of psychology*, vol. 15, no. 1, pp. 72–101, 1904.

[14] O. Levy, Y. Goldberg, and I. Dagan, "Improving distributional similarity with lessons learned from word embeddings," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 211–225, 2015.

[15] J. H. Lau and T. Baldwin, "An empirical evaluation of doc2vec with practical insights into document embedding generation," in *Proceedings of the 1st Workshop on Representation Learning for NLP*, 2016, pp. 78–86.

[16] M. Faruqui, Y. Tsvetkov, P. Rastogi, and C. Dyer, "Problems with evaluation of word embeddings using word similarity tasks," in *Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP*, 2016.

[17] A. Drachen, R. Sifa, and C. Thurau, "The Name In the Game: Patterns in Character Names and Gamer Tags," *Entertainment Computing*, vol. 5, no. 1, pp. 21–32, 2014.

[18] C. Ojeda, K. Cvejoski, R. Sifa, and C. Bauckhage, "Inverse Dynamical Inheritance in Stack Exchange Taxonomies," in *Proc. of AAAI AIIDE*, 2017.

User/Customer-centric Data Analytics

Feature Extraction and Large Activity-Set Recognition Using Mobile Phone Sensors

Wassim El Hajj, Cynthia El-Hayek
Computer Sc. Dpt.
American Univ. of Beirut
Beirut, Lebanon

Ghassen Ben Brahim
Computer Sc. Dpt.
Prince Mohammed Univ.
Alkhobar, Saudi Arabia

Hazem Hajj
Elec. And Computer Eng.
American Univ. of Beirut
Beirut, Lebanon

***Abstract*—In this work, the problem of activity recognition using data collected from the user's mobile phone is being addressed. We start with reviewing and discussing the limitations of most state of the art activity recognition approaches for mobile phone devices. Then, we present our approach recognizing a large set of activities that are comprehensive enough to cover most activities users engage in. Moreover, multiple environments are supported, for instance, home, work, and outdoors. Our approach suggests a single-level classification model that is accurate in terms of activity classification, comprehensive in terms of the large number of activities being covered, and applicable in the sense that it can be used in real settings. In the literature, these three properties are not existent altogether in a single approach. Existing approaches normally optimize their models for either one or, at a maximum two of the following properties: accuracy, comprehensiveness and applicability. Our results demonstrate that our approach achieves acceptable performance in terms of accuracy on a realistic dataset despite the significantly higher number of activities compared to the state-of-the-art activity recognition based models.**

Index Terms— Mobile Phone Sensing, Human Activity Recognition, Semantic Activity, Classification, Machine Learning.

I. INTRODUCTION

The recognition of human activities is becoming of high importance within several fields (medical, security, military applications, etc.) In a general setup, human activities are provided to a recommender system for processing and for recommending set of actions to be taken by the person under test. In the medical field for instance, recognizing activities such as walking, running, or cycling for a patient with diabetes becomes quite useful to provide feedback to the recommender system (caregiver in this case) about the patient's behavior for a set of actions to be taken.

Another example where activity recognition is very useful consists of improving the level of efficiency and performance of people when dong their regular daily life duties. According to [1], high stress level is the cause of almost a 100 million of lost workdays. It is also related to nearly 50% to 75% of diseases which can affect employees' performance, motivation towards goal achievements, low productivity, and could also result in several fatal health problems (physical, psychological, cardiovascular, etc.) One cause of stress for employees, for instance on their way to work, might involve being stuck in traffic. By being able to detect the actual situation or activity (stuck in traffic in this case), a recommender system can suggest some relaxing music that will eventually help leverage the stress level, and allow for better start of the day.

Accurately detecting Human Activities has become possible due to the advances in Mobile devices technology which became increasingly sophisticated. Cell phones now incorporates diverse types of sensors. These sensors include GPS sensors, vision sensors (cameras), audio sensors (microphones), acceleration sensors (accelerometers), etc. The availability of these sensors in mass-marketed communication devices creates exciting new opportunities for the advance in activity recognition related research.

In this respect, activity recognition has driven this research work since it constitutes significant information that will aid in enhancing a recommender system's suggestions. Our ultimate aim is to incorporate activity recognition within recommender systems to better personalize and enhance recommendations. We focus on activity recognition as it directly affects and allows the recommender system to suggest actions that aim towards enhancing the user's performance, efficiency, and eventually current state. Consequently, we focus on recognizing a large information set collected from user mobile phone comprising of sixteen activities: working regular, in a meeting, driving normally, stuck in traffic, in a vehicle, taking a break, eating, relaxing, watching TV, listening to music, playing games, exercising, biking, walking, running, spending time with family/friends.

The paper is organized as follows: Section II presents an extensive overview on existing mobile phone activity recognition approaches. Sections III presents our activity recognition process and a detailed description of our features extraction method. Section IV describes our experiment setup and results obtained by applying well-known classification schemes to the constructed dataset. Finally, conclusions and future work are presented in section V.

II. MOBILE PHONE BASED ACTIVITY RECOGNITION: OVERVIEW

Mobile phones with their increasing computational and sensing capabilities, storage and variety of applications have become an essential need in everyday user's life and events. The use of mobile phones to detect user's activity contributes in the development of healthcare monitoring systems, life logging applications and recommender systems. The recent approaches to the activity recognition problem focused on the use of mobile phone sensors to detect the user's daily activity.

In [2], the authors proposed a crowdsourcing framework that combines scene, event and phone context to recognize audio scenes (car, hall, indoor, restaurant, street) and events (keyboard, music, radio, speech, tv, walk, none) and phone

P. Haber et al. (Hrsg.), *Data Science – Analytics and Applications*, http://doi.org/10.1007/978-3-658-19287-7_9

context (in-pocket, out-pocket). The framework gathers everyday sounds from people and shares the audio models through a central cloud server. MFCC features were extracted from each audio clip and used to build a GMM. Each audio clip was represented by a histogram using the Gaussian Mixture Model (GMM). KNN algorithm was used as a classification tool to recognize a new audio clip and label it by a scene, an event and a phone context (in-pocket/out-pocket). The accuracy achieved was between 77.6 % and 88.9%. Although this approach achieved good accuracy, it recognizes scenes/states rather than actual semantic activities.

In [3], the authors proposed a continuous sensing engine for mobile phones that addresses the challenges of long term sensing and inference, and processing data from multiple sensors. The authors suggested a set of three pipelines: accelerometer, microphone and GPS to recognize user's activities and location. Each of the accelerometer and microphone was used separately to recognize a different set of activities and the GPS was used to detect the user's location. The accelerometer pipeline detects walking, cycling, running, vehicle and stationary activities and addresses the challenges of the phone body position errors and temporary states errors by using orientation-independent features and recognizing the transition states and the periods of interaction with the phone. The microphone pipeline detects brushing teeth, showering, typing, vacuuming, washing hands, crowd noise and street noise states and addresses the challenges of the resource efficiency by regulating the amount of data that enters the pipeline by filtering audio frames that contains speech and minimizing the redundant classification operations. After comparing multiple classifiers the decision tree was chosen for the accelerometer pipeline which achieved an accuracy of 94.52%. A GMM classifier was used for the audio-based classification and achieved an accuracy ranging from 0.6 to 0.98% depending on the activity and the decision tree classifier for recognizing voice frames achieved a recall of 85.35%. The GPS pipeline achieved an average error around of 45m for the different setup. This resource efficient and body position independent approach does not do any classification when voice is detected in the audio samples.

A hierarchical activity classification model for inferring semantic activities from accelerometer data only was proposed in [4]. The model was supported by the GPS sensor for location tagging. The semantic activities where referred to as Macro Activities and described as a sequence of smaller activities called Micro Activities. A lower layer detects micro-activities from the collected accelerometer data and an upper layer infers the semantic activities from the sequence of micro-activities. Statistical 2D and 3D features were extracted from the accelerometer data to detect micro activities and two feature extraction techniques, at the micro-activity level, were suggested and investigated. The list of micro-activities detected consists of: sit, sit-active, walk, loiter, bursty move, stand and using stairs. The semantic (macro) activities consist of office activities {O_work, O_break, O_coffee, O_toilet, O_meet, O_lunch} and home activities {H_work, H_relax, H_break, H_cook, H_eat, H_baby}. The lower layer was tested by a 10 fold cross validation approach using a set of classifiers (Decision tree, Naive Bayes, Bayesian Network, LibSVM and Adaboost) and an accuracy greater than 88% was achieved for all users. While this approach detects semantic, non-atomic activities, it is limited to only two locations: office and home and is not scalable in terms of activities covered since only movement based activities are being recognized using accelerometer data and sound based activities are ignored.

A hierarchical activity classification model was also introduced in [5] to detect a set of three activities: shopping, taking bus and moving (by walk). The model consists of a 2-Level HMM classifier where the first level detects a set of four actions (stand, walk, run, stair up/down) and the second level detects the actual activity where each activity consists of a sequence of actions. To recognize low-level actions, 3 different HMMs were trained for x, y and z accelerations respectively. For activity classification, an HMM was trained to detect the actual activity using the sequence of actions detected in level 1. The hierarchical HMM model was compared to 1 level HMM model and ANN Model and outperformed both in terms of average precision for the three activities. Though good performance results were achieved, this method suffers from the limited set of activities (3) to be recognized.

In [6], the authors describes “HARLib”, a human activity recognition library on the Android operating system. The authors used accelerometer built-in smartphone to recognize the user's activities. These activities include walking, running, sitting, standing, lying down and stairs. This models suffered from the limited number of activities considered as well as the low performance achieved in recognizing some of activities such as climbing the stairs and laying (62%).

Although these approaches achieved an acceptable level of accuracy in detecting activities, none of these approaches recognizes a large set of activities while combining accuracy, applicability and comprehensiveness, which will be the target of our mobile phone sensors based approach.

III. ACTIVITY RECOGNITION PROCESS

Our approach follows a typical activity recognition process with mobile phone sensors. Such process is made up of 5 main stages: (1) Raw Data Collection, (2) Data Filtering, (3) Data Processing, (4) Features extraction, and (5) Classification. In a more abstract way, the first 4 stages constitute the dataset construction phase and the last stage constitutes the Activity Recognition phase. Figure 1 captures the proposed activity recognition process. Next, we describe each of these blocks making up our process.

A. Raw Data Collection

To collect labeled data for training and performance evaluation purposes, we designed an Android data collection application designed and implemented on a Samsung GT-I9001 Android phone. The application operates in 2 modes: self-scanning mode and manual mode. The former is being triggered by a 1-minute timer and is responsible for collecting all sensors related data without the user intervention. These are the following:

- **Accelerometer data**. Collects X, Y and Z acceleration values for 8 seconds and goes inactive for the coming 12 seconds. According to [7] a duty cycle of 6 seconds

sampling and 10 seconds sleeping constitutes an acceptable trade-off between energy efficiency and robustness of state recognition. In our study we set the duty cycle to 8 seconds of sampling and 12 seconds of sleeping in order to obtain exactly 3 cycles of sampling for each one-minute interval of data collection.

- **Wi-Fi scan**. For energy efficiency reasons, we limited the data collection to one Wi-Fi scan every minute assuming that the location is subject to a minor change within a one minute interval in a location where Wi-Fi is available. A set of available Access Points alongside their Signal Strength and the Network ID is collected.
- **Audio recording**. For energy efficiency reasons and after set or trials we set the length of the audio recording to 10 sec. These are being collected at a certain period
- **GPS data**. Data about longitude and latitude is continuously collected.

The manual mode attempts to collect ground truth labels, which will, mainly serve as system training and performance evaluation purposes. In order to keep the user less engaged in the process of entering data, the time interval was set to 10 minutes, i.e. the user will have to enter the following set of information using dropdown menus every 10 minutes:

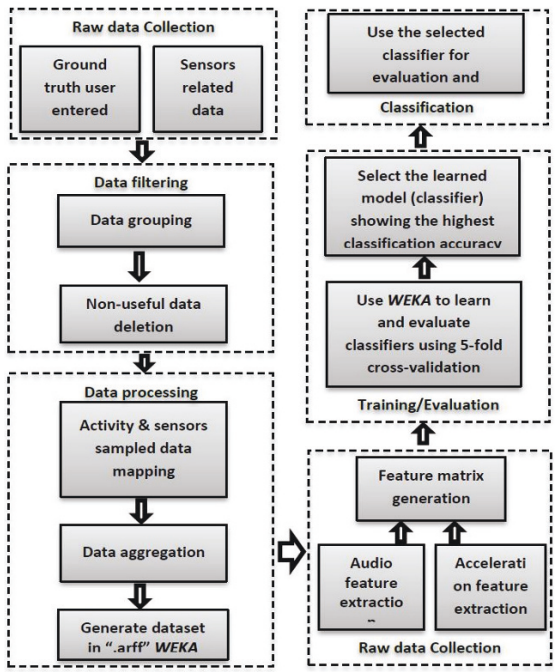


Figure 1. Proposed Activity Recognition Process

- **Location**. Where are you? The user needs to choose from the following: home, university, office, a client's office, restaurant, friend /relative's house, gym, street and other.
- **Activity**. What are you doing? working regular, in a meeting, driving normally, stuck in traffic, in a vehicle, taking a break, eating, relaxing, watching TV, listening to music, playing games, exercising, biking, walking, running and spending time with family/friends. Table 1 lists the set of activities along with their description.
- **Companion**. With how many people? The user needs to choose from the following: 0, 1, 2, 3, 4, 5 and more than 5.
- **Duration**. For how long? The user needs to choose from the following: 0 to 3 minutes, 4 to 7 minutes and more than 7 minutes.

At the end of this step, all data is logged into 5 different cvs files (4 corresponding to the sensors data and one for user entered data.) These will be filtered in the next step of data collection process

Table 1. Activity Description List

Activity	Description
Working regular	User in active regular daily work activities, excluding meetings.
In a meeting,	in a meeting with people
Driving normally	driving with continuous movement
Stuck in traffic	driving a car, or present in a car with non-continuous and minimal movement
In a vehicle	in a car with continuous movement, but not driving
Taking a break	User not performing any of his work tasks at the work place
Eating	having breakfast, lunch, or dinner
Relaxing	not performing any activity provided he is not located at his workplace
Watching TV	watching TV
Listening music	User's primary activity is listening to music
Playing games	User playing games
Exercising	User exercising, other than walking, running, or biking
Biking	user is biking
Walking	Walking: Outdoor location
Running	Running: Outdoor Location
Spending time with family/ friends	User with more than one person, chatting, and interacting with other people

B. Data Filtering

During this step, all 5 log files are transferred to an HP Pavillion DV61 station and are filtered using Java script

programs prior to the feature extraction step. Two main operations are performed on the raw data:

Step1. Sensor readings and ground truth labels grouping. The output of this step is to create a single structured file with time-based sequence of sensor reading events and activity annotations.

Step2. Non-useful data deletion. This step is mainly for structuring and deleting the non-needed data. The deletion process follows the following rules:

- Rule 1. DELETE Sensor data with no activity annotations as they do not contribute to the results in any way since they do not provide any information about the user's activity.
- Rule 2. DELETE Data sampled during the 10 minutes interval of an activity annotation but that do not correspond to the indicated duration of the activity.
- Rule 3. DELETE Sensor data corresponding to the user's interaction period with the phone while annotating his activity because the current activity might have been interrupted by the activity annotation process.

C. Data Processing

The input of this step is the single log file from the data filtering step. During this phase, further processing is being performed to allow easier manipulation of the huge amount of data by the feature extraction step. Two main operations are performed:

- **Mapping**. This step consists of mapping each of the sensors sampled data (collected every 1-minute interval) with the user current activity (collected every 10-minutes interval). The resulting file is a sequence of 1-minute sensor logs followed by their corresponding activity annotation.
- **Aggregation**. This step consists of aggregating all logs corresponding to a 1-minute interval in a single record. The single record consists of a timestamp, activity label, location, number of companions, duration of the activity, set of Wi-Fi APs with their signals strength and the name three data files. Each of these files hold the logs of a different sensor (accelerometer, audio and GPS) and the name of the data files holding the logs of a different sensor (accelerometer, audio and GPS).

D. Feature Extraction

In this section we describe the feature extraction process we applied to the final data set in order to extract audio and accelerometer features, as well as build the feature matrix that will be fed into our activity recognition model. These are described next.

- **Audio features extraction.** Audio features used consist of MFCC features, Spectral Rolloff, Spectral Flux and ZCR. To extract these features, a Unix-based tool called Yaafe was used which takes the audio files and the list of features to be extracted as input, and outputs a "csv" format file for each feature of an audio file. Each file contains the values of the different components of a single feature. These audio features together constitute the audio feature matrix.
- **Accelerometer features extraction**. Accelerometer features used in our approach consist of: Means, Variances, Mean-Magnitude, Magnitude-Mean, Single Magnitude Area (SMA) and Standard Deviation of Magnitude as detailed in the Table 2. In Table 2, x, y, and z corresponds to the standard 3 axes dimensions.
- **Feature matrix generation**. After extracting both audio and accelerometer features and building the corresponding matrices, they are combined into a single matrix to be fed into the classification tool, the rows of a matrix represent the combined feature vectors. The matrix is built by reading from both matrices and mapping each audio feature vector to the corresponding accelerometer feature vector.

Table 2: Accelerometer features

Mean	$\mathrm{AVG}(\sum x_i)$
Variance	$\mathrm{VAR}(\sum x_i)$
Mean-Magnitude	$\mathrm{AVG}(\sqrt{x_i^2 + y_i^2 + z_i^2})$
Magnitude-Mean	$\sqrt{\bar{x}_i^2 + \bar{y}_i^2 + \bar{z}_i^2}$
Single Magnitude Area	$\frac{1}{n}\sum_{i=1}^{n}(\lvert x_i\rvert + \lvert y_i\rvert + \lvert z_i\rvert)$
Standard Deviation of Magnitude	$\mathrm{STDEV}(\sqrt{x_i^2 + y_i^2 + z_i^2})$

- The result of the feature extraction step is a large matrix, the single element of which is the feature vector corresponding to each minute of data collection.

E. Classification

The Classification model used consists of a machine learning model - whose core is the SVM classifier which can be found in WEKA [8] as classifiers.functions.SMO. SVM was used following a set of preliminary experiments using Naïve Bayes, Decision Tree and SVM. In all cases, SVM has outperformed the previously mentioned classifier - which constitutes the reason behind using it in the proposed model.

The classification model takes the feature matrix that resulted from the audio and accelerometer features as input, and outputs a label relative to one out of the sixteen activities we are detecting in our work. The model was trained using a 5-Folds Cross Validation technique; a technique which segments the data set into 5 equal sets, and at each of the 5 iterations, takes 1 set as a test-set and the remaining 4 as training data.

In the next section, we describe the experimental set-up as well as performance results.

IV. Experimental Reults

A. Experiment Set-Up

Two users were engaged in this work, both carrying the application on a Samsung GT-I9001 Android phone over a period of six weeks. Both users carried the phones during their daily regular activities from home to the office then back home and during any other activity and in any location.

Tables 3 and 4 captures summary of the set of activities collected and recorded by both users. Both users had recorded instances for all activities, except for user 1 who recorded zero activity labeled "Playing Games" and user 2 who recorded zero activity labeled "Running".

After generating the feature matrix (as described in the previous section), WEKA data mining tool (version 3.7.13) [8], precisely the SVM classifier with a 5-Folds Cross Validation testing method were used. Classification results is presented in the next section.

Table 3. User 1 - Data Statistics

Activity	# of Instances
working regular	71
in a meeting	92
driving normally	20
stuck in traffic	83
in a vehicle	23
taking a break	17
eating	93
relaxing	13
watching tv	69
listening to music	76
playing games	0
exercising	14
biking	29
walking	100
running	18
spending time with family or friends	54
All Activities	**772**

Table 4. User 2 - Data Statistics

Activity	# of Instances
working regular	39
in a meeting	38
driving normally	28
stuck in traffic	5
in a vehicle	17
taking a break	8
eating	126
relaxing	28
watching tv	68
listening to music	9
playing games	15
exercising	14
biking	34
walking	244
running	0
spending time with family or friends	43
All Activities	**716**

B. Results

To evaluate the proposed approach, we applied the classification model on both users and collected “F-measure” metric as well as statistics about “False positive/negative” occurrences. For both users, the minimum F-measure was zero, however, the maximum F-Measure for User 1 was 0.932 for the activity "in a vehicle", and for User 2 was 0.970 for the activity "watching TV". F-Measure metrics are captured in Tables 5 and 6.

An in-depth look at the confusion matrix (Tables 7 and 8) highlights that the activities generating the highest number of false positives/false negatives are: Working Regular, In a Meeting, and Spending Time with Family or Friends. All these activities, present all 3 aspects of a semantic activity: sound, motion, and social aspect - which makes these activities more complex and thus present more difficulty in accurately predicting each. User 2 in specific, has exceptionally shown a high number of false negatives, showing the activity "eating" classified as "working regular" or "in a meeting", possibly due to the fact that "eating" in our case has been collected at both locations of "home" and "work", as well as the nature of the work of User 2, which might have resulted in both activities being similar.

Both confusion matrices show two types of misclassification: one between activities which are semantically similar (working regular-in a meeting for both users) - the other between activities which semantically different (eating-walking in the case of user 1, and eating-meeting in the case of user 2).

Table 5. F-measure – User 1

Activity	F-Measure
working regular	0.623
in a meeting	0.667
driving normally	0.918
stuck in traffic	0.000
in a vehicle	0.417
taking a break	0.000
eating	0.696
relaxing	0.814
watching tv	0.970
listening to music	0.941
playing games	0.815
exercising	0.692
biking	0.795
walking	0.966
running	-
spending time with family or friends	0.512
All Activities	**0.655**

Table 6. F-measure – User 2

Activity	F-Measure
working regular	0.717
in a meeting	0.654
driving normally	0.000
stuck in traffic	0.926
in a vehicle	0.936
taking a break	0.522
eating	0.747
relaxing	0.870
watching tv	0.831
listening to music	0.849
playing games	-
exercising	0.867
biking	0.881
walking	0.794
running	0.593
spending time with family or friends	0.578
All Activities	**0.718**

Table 7. Confusion Matrix - User 1

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	fp	fn
a: working regular	19	1	0	0	0	0	15	0	1	0	3	0	0	0	0	0	31	20
b: in a meeting	4	17	0	0	0	0	13	0	0	0	0	0	0	0	0	4	20	21
c: driving normally	0	0	26	0	0	0	2	0	0	0	0	0	0	0	0	0	1	2
d: stuck in traffic	0	0	0	4	0	0	0	0	1	0	0	0	0	0	0	0	0	1
e: in a vehicle	2	0	0	0	4	0	3	1	3	0	0	0	4	0	0	0	5	13
f: taking a break	4	0	0	0	0	0	3	0	0	0	0	0	0	0	0	1	1	8
g:eating	12	17	0	0	4	1	73	1	3	0	1	0	0	2	0	12	52	53
h: relaxing	0	0	0	0	0	0	0	21	2	0	4	1	0	0	0	0	6	7
i: watching tv	3	1	0	0	0	0	3	2	54	0	0	0	0	5	0	0	12	14
j: listening to music	3	0	0	0	0	0	0	1	0	4	1	0	0	0	0	0	0	5
k: playing games	1	0	0	0	0	0	4	1	0	0	9	0	0	0	0	0	10	6
l: exercising	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	1	0
m: biking	0	0	0	0	1	0	1	0	0	0	0	0	32	0	0	0	4	2
n: walking	0	0	1	0	0	0	0	0	0	0	0	0	0	243	0	0	14	1
o: unning	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p: spending time with family or friends	2	1	0	0	0	0	8	0	2	0	1	0	0	7	0	22	17	21

Table 8. Confusion Matrix - User 2

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	fp	fn
a: working regular	57	12	0	0	0	0	1	0	0	0	0	0	0	0	0	1	31	14
b: in a meeting	20	66	0	0	0	0	3	0	1	0	0	0	0	1	0	1	44	26
c: driving normally	0	0	13	5	1	0	0	0	0	0	0	0	1	0	0	0	4	7
d: stuck in traffic	0	0	1	81	0	0	1	0	0	0	0	0	0	0	0	0	11	2
e: in a vehicle	0	0	1	0	22	0	0	0	0	0	0	0	0	0	0	0	2	1
f: taking a break	2	4	2	0	0	6	0	0	0	0	0	2	0	0	0	1	0	11
g:eating	0	3	0	0	0	0	68	0	1	0	0	0	0	18	0	3	21	25
h: relaxing	0	3	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	3
i: watching tv	0	3	0	0	0	0	1	0	59	4	0	0	0	1	0	1	14	10
j: listening to music	3	3	0	0	0	0	1	0	9	59	0	0	0	0	0	1	4	17
k: playing games	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
l: exercising	0	0	0	0	0	0	0	0	0	0	0	13	1	0	0	0	3	1
m: biking	0	0	0	1	1	0	0	0	0	0	0	0	26	0	1	0	4	3
n: walking	2	5	0	0	0	0	8	0	0	0	0	0	0	83	0	2	26	17
o: unning	0	0	0	5	0	0	1	0	0	0	0	1	2	1	8	0	1	10
p: spending time with family or friends	4	11	0	0	0	0	5	0	3	0	0	0	0	5	0	26	10	28

V. CONCLUSIONS AND FUTURE WORK

In this research, we address the problem of activity recognition using data collected from the user's mobile phone. Our proposed approach suggests a single-level classification model that is accurate in terms of activity classification, comprehensive in terms of the large number of activities being covered, and applicable in the sense that it can be used in real settings. In literature, these three properties are not existent altogether in a single approach. Our results demonstrates that our approach achieves acceptable performance in terms of accuracy on a realistic dataset despite the significantly higher number of activities compared to the state-of-the-art activity recognition based models.

In future work, we propose improving the accuracy of our activity recognition model (i.e. reduce the number of misclassifications) by introducing a 2-level classifier approach. With this model we plan in grouping our large activity set into groups of activities that are semantically similar then have the algorithm run in 2 phases. The first phase attempts to reduce the misclassifications between semantically similar activities, and the second step attempts to reduce the misclassifications between semantically non-similar activities

REFERENCES

[1] Wang, Yi, et al. "A framework of energy efficient mobile sensing for automatic user state recognition." Proceedings of the 7th international conference on Mobile systems, applications, and services. ACM, 2009.

[2] Hwang, Kyuwoong, and Soo-Young Lee. "Environmental audio scene and activity recognition through mobile-based crowdsourcing." Consumer Electronics, IEEE Transactions on 58.2, 2012, 700-705.

[3] Lu, Hong, et al. "The Jigsaw continuous sensing engine for mobile phone applications." Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems. ACM, 2010.

[4] Yan, Zhixian, et al. "Semantic Activity Classification Using Locomotive Signatures from Mobile Phones.", 2012..

[5] Imtiaz, Subha, and Shakil Ahmad. "Impact Of Stress On Employee Productivity, Performance And Turnover; An Important Managerial Issue." International Review of Business Research Papers 5.4, 2009, 468-477.

[6] Young-Seol Lee and Sung-Bae Cho. "Activity Recognition Using Hierarchical Hidden Markov Models on a Smartphone with 3D Accelerometer." Hybrid Artificial Intelligent Systems,– Springer, 2011.

[7] H. C. Yang, Y. C. Li, Z. Y. Liu and J. Qiu, "HARLib: A human activity recognition library on Android," Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 11th International Computer Conference on, Chengdu, 2014, pp. 313-315.

[8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten: “The WEKA data mining software : an update”, SIGKDD Explorations, V. 11, number 1, 2009, Pages 10-18.

The Choice of Metric for Clustering of Electrical Power Distribution Consumers

Nikola Obrenović, Goran Vidaković
Schneider Electric DMS NS
Novi Sad, Serbia

Ivan Luković
Faculty of Technical Sciences
University of Novi Sad
Novi Sad, Serbia

***Abstract* — An important part of any power distribution management system data model is a model of load type. A load type represents typical load behaviour of a group of similar consumers, e.g. a group of residential, industrial or commercial consumers. A common method for creation of load types is the clustering of individual energy consumers based on their yearly consumption behaviour. To reach the satisfactory level of load type quality, the crucial decision is a choice of proper clustering similarity measure. In this paper, we present a comparison of different metrics, used as similarity measures in our process of load type creation. Additionally, we present a novel metric, also included in the comparison. The metrics and the quality of load types created therewith are assessed by using a real data set obtained from the distribution network smart meters.**

Keywords— Power load type creation; distribution consumers clustering; time series clustering; metric selection; high-dimensional data

I. Introduction

A precise model of consumer load is a prerequisite for obtaining correct results of distribution network state estimation, load forecasting and network planning calculations [1, 2]. Since the state estimation calculation represents one of the basic functionalities of a power Distribution Management System (DMS), as a prerequisite for other calculations, the consumer load model is essential for the functioning of the whole system.

Active (P) and reactive (Q) power can be measured on all or just on a selected set of distribution consumers. However, raw measurements of separate consumers' consumption are a poor representation of the network load due to the high variations in single consumer's behaviour throughout a year or between years [2]. On the other hand, if similar consumers are grouped, we obtain a statistically better representation of the load at the level of the whole group and a smoother load curve, that can be adequately used for distribution network calculations and forecasting [2].

Therefore, in a model of power distribution network, the load of an electrical consumer is represented with the following elements:

- annual average active power P_{avg},
- annual average reactive power Q_{avg}, and
- a load type.

The load type represents consumption behaviour of a group of similar consumers and consists of a set of relative load profiles. Each relative profile, or profile for short, represents consumption of active or reactive power within 24 hours, for one season and one day type, given in relative units. The commonly used seasons are: spring, summer, autumn and winter, while common day types are: Working Day, Saturday, and Sunday-Holiday. The number of load profiles in each load type is the product of the number of seasons, the number of day types and the number of observed physical quantities. Consequently, the number of load profiles in each load type is typically 24, as we also adopt in this research.

A load profile typically consists of 96 points, where each point is a relative value at a quarter hour, starting from the midnight (Figure 1). The actual active (resp. reactive) power for a particular consumer and at a particular time is determined by:

1) selecting the appropriate load type profile of active (resp. reactive) power, for the season and day type that correspond to the observed date,
2) reading the profile value, based on the observed time, and
3) multiplying the read value with P_{avg} (resp. Q_{avg}).

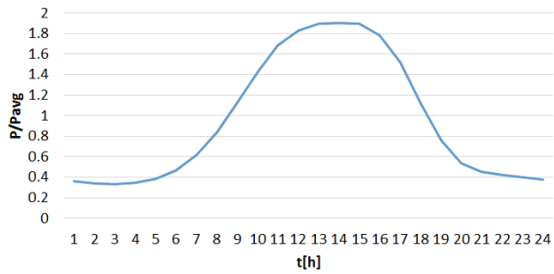


Fig. 1. Relative daily profile of active power

To obtain the described load type profiles, for each consumer we create a set of load profiles for the same combinations of seasons and day types as for the load types. I.e. each consumer is represented with the set of load profiles, and profiles' values determine features to be used for grouping the similar consumers. For this task, there are many algorithms tested and applied, such as described in [1, 2, 3]. In our approach, we use k-Means++ clustering algorithm [4]. However, as each clustered consumer is usually described with 24 load profiles, each of which has 96 values, a consumer is described with a very large

P. Haber et al. (Hrsg.), *Data Science – Analytics and Applications*, http://doi.org/10.1007/978-3-658-19287-7_10

total number of dimensions, 24*96=2.304. Consequently, our clustering problem is burdened with the dimensionality curse.

In order to address this issue, we have evaluated several metrics which can be used with the k-Means++ algorithm, with the purpose to identify a metric that provides the best clustering results in our case. In this paper, we present the evaluation results. Also, we have created an original metric, named Curve Shape Distance, and included it in the evaluation. In this paper, we also present Curve Shape Distance, along with the proof that the proposed function satisfies all properties of a metric. To the best of our knowledge, the developed metric has not been described in the available literature.

Apart from Introduction and Conclusion, the paper is organized as follows. In Section 2 we present methods already applied for clustering electrical power consumers and determining load types. In Section 3 we present our implementation of load type creation algorithm. The evaluated metrics are described in detail in Section 4, followed by the metrics evaluation results, given in Section 5.

II. Related Work

A vast number of approaches considering the problem of determining groups of similar power consumers, i.e. determining characteristic load types, can be found. The different approaches propose different clustering and classification algorithms. However, to the best of our knowledge, none of them gives a special attention to the selection of the best metric to be used in their approach. In the following text, we present several of the analysed approaches.

In [1], authors propose the usage of the ISODATA algorithm and the Euclidean metric, for the purpose of clustering power consumers' load data. Thereby, a single consumer load is represented with a normalized load pattern vector containing 2020 dimensions, which is similar to our approach. In order to justify the approach, the authors compare the obtained results with the results obtained with k-Means algorithm. However, the choice of the Euclidean metric is not justified in the paper, nor any other metric is assessed.

In [3], authors compare several unsupervised clustering algorithms, i.e. modified follow-the-leader, hierarchical clustering, K-means, fuzzy K-means, and the self-organizing maps, in order to find the best solution for grouping customers with similar electrical behaviour. Additionally, authors assess different dimensionality reduction techniques, e.g. Sammon map, principal component analysis, and curvilinear component analysis, for the same purpose. Thereby, the Euclidean metric and its variants are used with all compared algorithms and a set of validity indices is used for comparison of the clustering results. In this paper, we also use validity indices for assessment of resulting clusters. On the other hand, authors in [3] have not analysed any alternative metric and consequently, the influence of the metric on the clustering results quality, which is the main objective of our research presented in this paper.

In [2], Chicco gives an overview of different clustering technics used for load pattern creation, e.g. k-Means, Follow-the-leader, Self-organizing maps, Support vector clustering, Probabilistic neural network, etc. The author states that different metrics can be used in the clustering validity assessment. However, the metrics that can be used with the discussed algorithms have not been analysed. Such task has been just declared in the paper as "a challenging aspect".

In [5], the authors assessed several metrics for the purpose of clustering energy consumption daily profiles of university buildings, collected over a period of 4 months. Thereby, the authors made an assumption that a correlation-based metric would yield the best clustering results. However, the test showed that the Euclidean metric outperformed the correlation metric.

In contrast to [5], we process data from distribution consumers of various types simultaneously, and clustered consumption data span the whole year. Since the selection of a metric, i.e. a similarity measure, can be dependent of the nature of the clustered time series, we could not directly apply conclusions given in [5] and had to evaluate the metrics on our data. However, in our tests, we have obtained similar results regarding the Euclidean and the correlation metrics, as in [5].

III. Load Type Creation Algorithm

Input data for the presented algorithm are 15-minute measurements of P and Q per consumer, for each day in a period of one year. On the other hand, load type profiles are aggregated at the level of season and day type in order to decrease the stochastic component of load behaviour as much as possible. The commonly selected seasons are winter, spring, summer, and autumn, while the usual day types are: Working Day, Saturday, and Sunday with Holidays. These seasons and day types have been used in the case of this evaluation as well.

In order to obtain the described load groups, the algorithm consists of the following steps, abbreviatedly presented in Figure 2.

Step 1. Within each consumer, we calculate from the input measurements, an average daily load profile (DLP) for each different combination of physical quantity (P or Q), season and day type. Thereby, we obtain a set of 24 DLPs for each consumer.

Step 2. In order for a load type to represent consumers of different power magnitude, consumer's DLPs are divided by P_{avg}, in the case of DLPs of P, or by Q_{avg}, in the case of Q DLPs. The result is a set of 24 relative DLPs (RDLPs) per each consumer.

Step 3. All RDLPs of one consumer are chained into one long array of values. The order of RDLPs in each consumer must be the same so that each index in the resulting array denotes the same physical quantity, season, day type and the ordinal number of a measurement in a day, for each consumer. The obtained array of RDLP values represents a data instance that describes consumption behaviour of a single consumer throughout a year. The number of dimensions of each consumer is number of physical quantities times the number of seasons times the number of day types, which is 2.304 in our case.

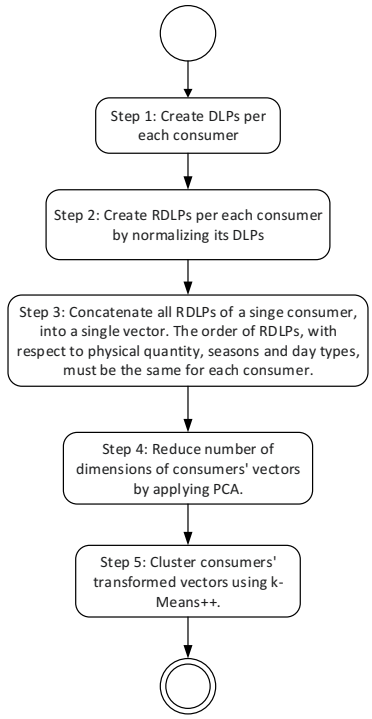


Fig. 2. Load Type Creation Algorithm

Step 4. Further, we deploy on our data the dimensionality reduction method called Principal Component Analysis (PCA, [4]) and decrease the number of dimensions of each consumer to 36. The number of targeted dimensions has been empirically determined, by a visual inspection of the obtained clusters, and its selection could be a matter of future research.

This step is required in order to reduce algorithm execution time over untransformed data and to avoid the dimensionality curse. I.e., to the best of our knowledge, the existing clustering algorithms and metrics usually cannot yield meaningful clusters from data of such high dimensionality.

Step 5. The reduced data instances, i.e. consumers are further clustered using the k-Means++ algorithm with one of the metrics described in the following section.

Each resulting cluster represents one load type, while the cluster centroid represents the load type profiles.

IV. Analyzed Similarity Measures

Euclidean distance is a commonly used metric for clustering data which attributes are not correlated or where this correlation can be ignored. Euclidean distance between data points x and y in an n-dimensional space is given with the following formula:

$$d(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}, \quad (1)$$

where x_k and y_k, $k \in \{1,..,n\}$, are data point dimensions.

However, to the best of our knowledge, there is no proof that the k-Means++ algorithm provides the best clustering results by using the Euclidean metric, at least not in the case of distribution load profiles clustering. Hence, we have analysed and tested several other metrics in order to find a metric that would provide results of better quality.

Additionally, energy consumption depends highly on factors shared by all consumers, such as weather conditions, day period, day type, season, etc. Therefore, in the case of consumer load meter readings, arrays of RDLPs of different consumers might be correlated to some extent. Therefore, we expected that correlation-based similarity measures in particular would provide an improvement in results quality. The analysed metrics are presented in the following subsections.

A. Manhattan distance

The Minkowski distance [4] is a generalization of the Euclidean distance, given with the following formula:

$$d(x,y) = \left(\sum_{k=1}^{n}|x_k - y_k|^r\right)^{\frac{1}{r}}, \quad (2)$$

where r is a parameter, n is the number of dimensions, while x_k and y_k are dimensions of data points x and y. The Manhattan, or L1, distance is obtained by settings the parameter r to 1.

B. Cosine similarity

Cosine similarity [4] is a similarity measure based on cosine of the angle between two vectors. Cosine similarity is usually used for the positive cones, where dimensions of all vectors are positive. This requirement is satisfied in our case. The value of the cosine similarity is always in [0,1], regardless of the number of dimensions, making it suitable for usage in highly dimensional vector spaces. The cosine similarity is calculated by the formula:

$$s(x,y) = cos(x,y) = \frac{x \cdot y}{\|x\|\|y\|}, \quad (3)$$

where $x \cdot y$ is scalar product of vectors x and y, while $\|x\|$ and $\|y\|$ are modulo of vectors x and y, respectively.

Cosine distance is a complement of the cosine similarity, calculated as:

$$d(x,y) = 1 - s(x,y). \quad (4)$$

C. Cross Correlation

Cross Correlation [6] represents similarity measure between two wave signals. If assumed that $x(i)$ and $y(i)$, i=0,...,n, are two time series, the Cross Correlation r with delay d is defined as:

$$r(x,y,d) = \frac{\sum_i[(x(i)-mx)\cdot(y(i-d)-my)]}{\sqrt{\sum_i(x(i)-mx)^2}\cdot\sqrt{\sum_i(y(i-d)-my)^2}}, \quad (5)$$

where mx and my represent the average values of the time series $x(i)$ and $y(i)$, respectively. Value of r is between –1 and 1, where 1 denotes the maximum correlation, 0 the absence of correlation, and –1 the maximum inverse correlation. In the case of measured active and reactive power, d is equal to 0, since all measurements are taken at the same timestamps. Furthermore, since Cross Correlation measures similarity, we have used its complement:

$$d(x,y) = 1 - r(x,y,0). \quad (6)$$

D. Spearman's rank correlation coefficient

Spearman's rank correlation coefficient [7] first ranks each value in the time series in the following way. The variable with the smallest value is assigned rank 1, the variable with the second smallest value, rank 2, and so forth. By this, we normalize all time series values to the same, unitless domain. Afterwards, Spearman's coefficient is calculated as:

$$\rho = 1 - \left(\frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}\right), \quad (7)$$

where d_i is difference of ranks of the time series values x_i and y_i, and n is dimensionality of the time series.

E. Curve Shape Distance

This metrics represents an extension of the Euclidean metric, with the linear approximation of the first derivate in each point of the time series. This metric has been originally defined during the evaluation of existing metrics and, to the best of our knowledge, cannot be found elsewhere in the literature. The idea of this metric is grounded in the fact that two curves, i.e. time series, are more similar if they have more similar first derivate in each point of theirs, in addition to the similarity of the series values.

The metric is defined by the following formula:

$$d(x,y) = E(x,y) + \sum_{k=1}^{n-1} |(x_{k+1} - x_k) - (y_{k+1} - y_k)|/\Delta t, \quad (8)$$

where E is the Euclidean distance, n is the number of time series dimensions, x_k and y_k are k^{th} elements of time series x and y, respectively, and Δt is the difference on x-axis between two adjacent elements of the time series. In the case of untransformed data, Δt represents the time difference between two subsequent measurements of the time series, in our case a quarter hour. On the other hand, the x-axis of the PCA transformed time series represents ordinal numbers of the transformed dimensions. Hence, Δt represents in this case the difference between ordinal numbers of two adjacent dimensions and it is always equal to 1.

In both cases, Δt is a constant and positive value for all clustered time series. This property of Δt is important for proving that Curve Shape Distance is a metric. The proof is presented in the rest of this section.

Proof. In order for function $d(x,y)$ to be a metric, it must satisfy the following four properties [4]:

1. *non-negativity*: $d(x,y) \geq 0$;
2. *identity of indiscernibles*: $d(x,y) = 0 \Leftrightarrow x = y$;
3. *symmetry*: $d(x,y) = d(y,x)$; and
4. *triangle inequality*: $d(x,z) \leq d(x,y) + d(y,z)$.

In the following text, we prove that each of these properties stand for the Curve Shape Distance function. For the sake of brevity, we will denote the second addend of (8) with $D(x,y)$.

1* Non-negativity. Since $E(x,y)$ is the Euclidean metric, it is always non-negative. $D(x,y)$ represents the ratio between the sum of absolute values, which is always non-negative, and Δt, which is non-negative, too. Since both addends of (8) are non-negative, the whole (8) is non-negative, as well.

2* Identity of indiscernibles. (→) If (8) is equal to 0, then $E(x,y)$ is also equal to 0 since both addends are non-negative. Further, since $E(x,y)$ is the Euclidean metric and it is equal to 0, it yields that $x = y$. (←) If $x = y$, then $E(x,y) = 0$. Also, if $x = y$, then $x_i = y_i$, $i \in \{1,..,n\}$. This further yields that $(x_{k+1} - x_k) = (y_{k+1} - y_k)$, $k \in \{1,..,n-1\}$ and that $D(x,y) = 0$, too.

3* Symmetry. From the symmetry property of the Euclidean metric and the absolute value, the symmetry of the Curve Shape Distance metric stands as well.

4* Triangle inequality. From the triangle inequality of the absolute value function, it follows:

$$|(x_{k+1} - x_k) - (z_{k+1} - z_k)| \leq |(x_{k+1} - x_k) - (y_{k+1} - y_k)| + |(y_{k+1} - y_k) - (z_{k+1} - z_k)|, \quad k \in \{1,..n-1\}.$$

Since Δt is a constant and positive value, it further yields $D(x,z) \leq D(x,y) + D(y,z)$.

Finally, since the triangle inequality stands for the Euclidean metric and the second addend $D(x,y)$, it stands for the whole (8).

With 1*, 2*, 3* and 4*, we have proven that (8) is a metric. □

V. Experimental Results

A common method for clustering result evaluation is a validity index. The validity index is a formula whose value represents the clustering result quality and can be used to compare the quality of results obtained from the same algorithm run with different parameters or different algorithms [8].

For the purpose of comparing results of clustering with different metrics, we have used Davies-Bouldin (DB) [9] and SD validity [10] indices. The DB index measures the mean of similarity of each cluster to its most similar cluster. This measure can indicate how similar clusters are and if some of them should be merged. The SD index is based on average scatter of the clusters and total distance between the clusters. The clustering result is better if created clusters are less scattered and more distant one from each other [10]. The lower value of DB and SD validity indices means the better clustering result.

For testing, we used 3 data sets, each containing 2000 data instances, i.e. power distribution consumers from a European network. The data sets are denoted as CR1, CR2 and CR3. With each assessed metric, we ran k-Means++ clustering of each data set for 19 times, with targeted number of clusters ranging from 2 to 20 clusters, and chose the clustering configuration with the best validity index.

Since DB and SD validity indices also require a metric for their calculation, each obtained clustering result was assessed with the same metric that was used for its creation.

Among Minkowski metrics, Euclidean metric created between 11 and 14 clusters per data set, while Manhattan metric gave either 7 or 8 clusters depending on a data set. However, Manhattan metric clustered most data instances, more than 90%, into a single cluster which was not expected in the selected data set, nor generally in the case of electrical power consumption data. For this reason, we have disqualified the Minkowski metric from this evaluation.

With the Cosine similarity, Cross Correlation and Spearman's coefficient, clustering created 3 clusters per data

set. Curve Shape Distance created between 10 and 14 clusters per data set.

The DB validity index values obtained with the tested metrics are given in Table 1. In Table 2, the values of the SD index are given. The lowest index values per data set are given in bold letters.

The conclusion derived from the experimental values is that the Euclidean metrics gave the best results, followed by Curve Shape Distance.

TABLE I. DB VALIDITY INDEX VALUES WITH PCA

	CR1	CR2	CR3
Euclidean (L2)	**1.52**	**1.59**	**1.25**
Cosine	2.70	2.33	2.63
Cross Correlation	1.85	1.92	1.89
Spearman	4.35	4.76	4.76
Curve Shape Dis.	1.79	1.67	1.79

TABLE II. SD VALIDITY INDEX VALUES WITH PCA

	CR1	CR2	CR3
Euclidean (L2)	**0.70**	0.72	0.70
Cosine	1.00	1.02	0.80
Cross Correlation	1.00	1.18	0.96
Spearman	1.06	1.23	1.23
Curve Shape Dis.	0.84	**0.70**	**0.63**

Our initial assumption was that correlation metrics would give betters clusters than the others since the processed power data might have a certain amount of correlation. As this assumption proved wrong, we further assumed that correlation element may have been masked by the PCA transformation. Therefore, we repeated the test without dimensionality reduction with PCA. The corresponding results are given in Table 3, for DB index, and in Table 4, for SD index. The lowest index values per data set are again given in bold letters.

TABLE III. DB VALIDITY INDEX VALUES WITHOUT PCA

	CR1	CR2	CR3
Euclidean (L2)	1.56	**1.33**	1.43
Cosine	1.59	1.61	1.59
Cross Correlation	2.22	2.17	2.17
Spearman	2.38	2.44	2.56
Curve Shape Dis.	**1.28**	1.47	**1**

TABLE IV. SD VALIDITY INDEX VALUES WITHOUT PCA

	CR1	CR2	CR3
Euclidean (L2)	0.79	0.72	0.69
Cosine	0.68	0.62	0.65
Cross Correlation	1.32	1.39	1.16
Spearman	1.25	1.25	1.19
Curve Shape Dis.	**0.35**	**0.47**	**0.42**

In spite a priori expectations, removing the PCA transformation did not influence significantly to the clustering results with Cross and Spearman's correlation, while the indices obtained with the Euclidean metric were slightly higher. On the other hand, indices obtained with the Cosine similarity and Curve Shape Distance were lower without PCA, meaning that the clusters created in such manner were of a better quality. The best indices were obtained with Curve Shape Distance. Also, the effect of the dimensionality curse was not present in the resulting clusters obtained with this metric.

VI. CONCLUSION

A goal of this work has been to determine the best metric, from the set of available metrics, to be used with the k-Means++ algorithm, for the task of clustering electrical consumer load profiles. We have tested metrics against the original data and the same data transformed with PCA, which was used to reduce the number of dimensions. The initial assumption was that correlation-based metrics might yield the best clustering results, due to the possible correlation within the clustered data.

The conclusion of the experiments with the PCA transformed data is that the lowest validity indices, i.e. the clusters of the best quality, are obtained with the Euclidean metric. The second best metric was Curve Shape Distance, originally proposed in this paper. The correlation-based metrics provided significantly worse results than the Euclidean metric.

The reason of poor performance of the correlation-based metrics was assumed to be the PCA transformation, which could have removed the correlation between the data instances. Therefore, the tests have been repeated with the original, untransformed data.

This time, the best clustering results are obtained with Curve Shape Distance, while the Euclidean and the Cosine metrics could be both ranked as the second best. The Cross Correlation and Spearman's rank correlation metrics gave again unexpectedly high validity indices, i.e. the clusters of bad quality.

Further, the tests showed that validity indices with Curve Shape Distance metric applied on untransformed data are lower than validity indices when Euclidean metric is applied on the PCA transformed data. Consequently, we also conclude that if we have a low number of data instances or hardware powerful enough to run clustering on the original data, we should opt for that approach and the Curve Shape Distance metric.

As part of our further research, we plan to assess other clustering algorithms, such as Follow-the-leader or an agglomerative or fuzzy clustering algorithm, and the quality of load types obtained therewith. Along with this, a complementary research path would be development of a consumption-based metric, which will account the physical properties of the clustered load profiles. We believe that such a tailored metric might improve clustering results. By that, we are working towards creating the best available solution for clustering the power consumption data and creation of high-quality load types, which are of great importance for forecasting the power consumption and estimation of power grid state.

Finally, we plan to repeat assessment of the metrics on a larger number of data sets, which should also be greater in size. By this, we will provide possibility for further analysis of the statistical significance of the experimental results and provide a firm confirmation of the experiment conclusions, presented in this paper.

REFERENCES

[1] A. Mutanen, M. Ruska, S. Repo, and P. J□rventausta, "Consumer Classification and Load Profiling Method for Distribution Systems," IEEE Transactions on Power Delivery, vol. 26, pp. 1755-1763, July 2011.

[2] G. Chicco, "Clustering Methods for Electrical Load Pattern Classification," Scientific Bulletin of the Electrical Engineering Faculty, Year 2010, No. 3 (14), ISSN 1843-6188, pp. 5-13.

[3] G. Chicco, R. Napoli, and F. Piglione, "Comparisons among clustering techniques for electricity customer classification," IEEE Transactions on Power Systems, vol. 21, pp. 933-940, May 2006.

[4] P. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, 2005.

[5] F. Iglesias and W. Kastner, "Analysis of Similarity Measures in Times Series Clustering for the Discovery of Building Energy Patterns", *Energies* vol.6, pp. 579-597, January 2013.

[6] P. Bourke, "Cross Correlation," August 1996, http://paulbourke.net/miscellaneous/correlate, accessed August 2016.

[7] J. Black, N. Hashimzade, and G. Myles, A Dictionary of Economics (3 ed.). Oxford University Press, 2009.

[8] F. Kovács, C. Legány and A. Babos, "Cluster Validity Measurement Techniques," Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED©06)pp. 388-393, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 2006.

[9] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 1, pp. 224-227, April 1979.

[10] M. Halkidi and M. Vazirgiannis, and Y. Batistakis, "Quality Scheme Assessment in the Clustering Process," Proc. of the 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD ©00) pp. 265-276, Springer-Verlag, London, UK, 2000.

Evolution of the Bitcoin Address Graph

An Exploratory Longitudinal Study

Erwin Filtz, Axel Polleres
Institute for Information Business
Vienna University of Economics and Business &
Complexity Science Hub Vienna
Vienna, Austria

Roman Karl, Bernhard Haslhofer
Digital Insight Lab
Austrian Institute of Technology
Vienna, Austria

Abstract - **Bitcoin is a decentralized virtual currency, which can be used to execute pseudo-anonymous payments globally within a short period of time and comparably low transaction costs. In this paper, we present initial results of a longitudinal study conducted over the Bitcoin address graph, which contains all addresses and transactions from the beginning of Bitcoin in January 2009 until 31st of August 2016. Our analysis reveals a highly-skewed degree distribution with a small number of outliers and illustrates that the entire graph is expanding rapidly. Furthermore, it demonstrates the power of address clustering heuristics for identifying real-world actors, who prefer to use Bitcoin for transferring rather than storing value. We believe that this paper provides novel insight into virtual currency ecosystems, which can inform the design of future analytics methods and infrastructures.**

Keywords—Bitcoin, network analytics, virtual currencies

I. INTRODUCTION

Bitcoin [1] is the most prominent representative of decentralized, unregulated virtual currencies, which are based on cryptographic technologies – also known as "cryptocurrencies". In contrast to other fiat currencies (e.g. EUR, USD), such currencies have no pre-assumed identities, are not controlled by any central authorities, but are organized as a peer-to-peer network. Furthermore, all executed transactions are stored in a public, distributed ledger called the Bitcoin *blockchain*.

While the public ledger provides a high level of transparency on past transactions, it does not explicitly reveal details about real-world actors involved as senders or receivers of financial transactions. A single *transaction* is represented by a list of inputs pointing back to outputs of previous transactions and a list of outputs, each reflecting a certain Bitcoin value that has been transferred to some specific recipient's *address*. A Bitcoin address is an alphanumeric string derived from the public key of an asymmetric key pair generated by some Bitcoin user. Every user can hold multiple key-pairs (and addresses) in a so-called "*wallet*", and is encouraged to use a new address for each transaction to increase the level of anonymity.

The design of Bitcoin implies that the balance of an address is not stored explicitly in the blockchain but must be calculated by summing up all unspent outputs associated with that address. Additionally, a Bitcoin value associated with an address in an output cannot be partially spent; and the sum of inputs must be equal to the sum of outputs in each transaction. It is, however, possible to transfer input values exceeding the outputs ("*change*") back to the same address or to another address owned by the same real-world actor.

The goal of this paper is to present initial results of a longitudinal study conducted over the Bitcoin address graph as a Data Science use case. Our contributions can be summarized as follows:

- We provided a comprehensive graph representation of all Bitcoin addresses and transactions from the beginning of its existence (2009-01-03) until the time of this writing (2016-10-31).
- We conducted a structural analysis of the address graph investigating the change in the structure of the graph over time.
- We investigated the fraction Bitcoin addresses that can explicitly or implicitly be assigned to real-world actors and show how they change over time.
- We examined the transaction behavior of users, taking into consideration exchange rates between virtual and fiat currencies.
- We analyzed the function of virtual currencies from a user perspective, by analyzing the activity periods of addresses and address clusters.

A potential practical use of presented methods and results lies in the implementation in real-time virtual currency analytics platforms, which could provide insight into the current state and evolution of virtual currency ecosystems, such as Bitcoin.

The remainder of this paper is organized as follows: Section II provides an overview of related research in the field of virtual currency analytics. Section III outlines the core concepts of the Bitcoin system. Section IV introduces the dataset we used and the analyses we conducted, as well as the initial results of our investigations.

P. Haber et al. (Hrsg.), *Data Science – Analytics and Applications*, http://doi.org/10.1007/978-3-658-19287-7_11

II. Related Work

A strong focus of previous research has been on the anonymity property of Bitcoin and possible strategies for de-anonymizing addresses. This is motivated by the strong association between virtual currencies and cybercrime (e.g., ransomware, DDoS attacks). Currently, there is discussion about whether authorities should force "*wallet providers […] to apply customer due diligence controls, ending the anonymity associated with such exchanges*."[1]

The following research investigated relevant Bitcoin anonymity issues. Ron and Shamir [2] analyzed the typical behavior of Bitcoin users and how they act to obfuscate the flow of Bitcoins to remain anonymous. Meiklejohn et al. [3] tried to reveal real-world identities of users by heuristic clustering and re-identification attacks. Biryukov et al. [4] described linking the IP address of transactions to user pseudonyms even when they are behind NATs or firewalls.. Monaco [5] explained de-anonymization by identifying users based on their behavior. He found that the transaction behavior of users is nonrandom and nonlinear and that users follow the same behavioral patterns in the long run. Fleder et al. [6] applied web scraping and transaction fingerprinting to reveal the identity of real-world actors.

Other research investigated the properties and behaviors of real-world actors in the Bitcoin ecosystem. M□ser et al. [7] analyzed the reliability of *mixing services*, which can be used to camouflage transactions by breaking the connection between a Bitcoin address sending coins and the address(s) they are sent to. They concluded that there are quality differences in existing services. Reid and Harrigan [8] analyzed an alleged theft of Bitcoins from a Bitcoin *exchange*.

Graph representations extracted from the Bitcoin blockchain were also studied: Holtz et al. [9] investigated the properties of the Bitcoin graph around the announcement of a Bitcoin gaming site by splitting the graph into small parts and comparing certain properties before and after the launch of the gaming site. Miller et al. [10] revealed deeper insight into the Bitcoin topology, the broadcast method, and the role of influential nodes taking advantage over other nodes. Harrigan and Fretter [11] addressed the advantages, disadvantages and effectiveness of a commonly used address clustering heuristic that allocates all input addresses of a transaction to the same real-world actor.. Another study by Kondor et al. [12] analyzed the structure of the transaction network and the evolution of wealth in this network. Further analysis of the bitcoin transaction network is conducted by Ober et al. [13] this investigation focused on global properties of the Bitcoin graph with the result that several parameters remained steady over the last 1.5 years. However, a systematic analysis of the Bitcoin graph and its evolution over time has not yet attracted great attention. Haslhofer et al. [14] and Spagnuolo et al. [15] provided virtual currency analytics tools to implement some of the methods mentioned.

[1] http://www.consilium.europa.eu/en/press/press-releases/2016/12/20-money-laundering-and-terrorist-financing/

III. Bitcoin

A. Basic Entities

The Bitcoin system and its working principle with *addresses*, *wallets* and *transactions* was first described by Nakamoto in 2008 [1]. The basic structure and the relations of its elements is illustrated in Figure 1, which shows the relationship between *addresses*, *wallets*, *transactions*, and (real-world) *users* in the Bitcoin ecosystem. Each user U can have zero to multiple addresses A. In this example, we have two users U_1 and U_2, who hold two addresses in their respective wallets: U_1 holds addresses A_1 and A_2 and user U_2 addresses A_3 and A_4. Each (non-coinbase) transaction T then links at least two, but typically three, and up to an arbitrary number of addresses, in which the value of each transaction must be at least 1 Satoshi, which is 10^{-8} Bitcoins (BTC).

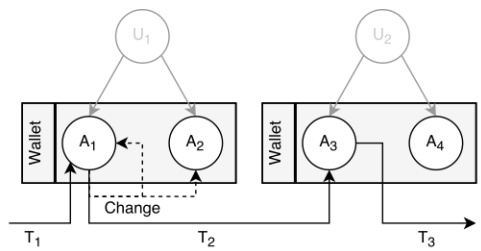


Fig. 1. Schematic view of the Bitcoin system

In Figure 1 let us assume that U_1 received 1 BTC from another user in transaction T_1 and wants to pay U_2 0.75 BTC due to a contractual obligation. As mentioned earlier, U_1 cannot transfer 0.75 BTC to U_2 directly. Instead, U_1 must spend the entire amount of 1 BTC associated with A_1 in T_2, specifying that 0.75 BTC goes to A_3 held by U_2. The remaining 0.25 BTC are *change* and must be transferred to another address, which can be A_1 or a newly generated address A_2 held by U_1. It is an assumption that the addresses A_1 and A_2 belong to user U_1. The only fact that can be infered from the blockchain is that there were 0.75 BTC sent from address A_1 to address A_3 in transaction T_2 and 0.25 BTC from A_1 to A_1 or A_2 as *change*.

Newly generated transactions are broadcasted to the Bitcoin peer-to-peer network and collected by special-purpose nodes, the so-called *miners* that try to combine them into a new *block*, which is issued approximately every 10 minutes and added to the blockchain with a timestamp resulting in a monotonously growing, temporarily ordered transaction sequence. Mining is a competitive and highly resource-intensive task (*proof-of-work*) and follows a pre-defined consensus protocol; which are both beyond the scope of this paper.

B. Constructing the Bitcoin Address Graph

The Bitcoin address graph can be constructed by extracting all transactions from the blockchain and creating a *property graph*, in which each node represents an address and each edge a transaction that has taken place between a source and a target address.

Each node (address) and edge (transaction) can carry additional descriptive properties: typical properties for

addresses are *tags* providing additional contextual information about an address. Such tags might be collected by crawling the Web. Possible properties for edges are the number of transactions or the flow of Bitcoins between two addresses.

In order to quantify the flow of Bitcoins between two addresses, it must be understood that unlike the real-world banking system, a Bitcoin transaction represents an m:n relationship between addresses. Thus, a transaction can have multiple input and multiple output addresses, as illustrated in Figure 2. In this case we assume that there is a transaction T having addresses A_1 with a value of 2 BTC and A_2 with a value of 5 BTC as an input. The outputs of T are addresses A_3 receiving 3 BTC and A_4 receiving 4 BTC. In Bitcoin, it is impossible to assign a specific value of an input to a specific output address. Even though A_3 receives 3 BTC, the source of the 3 BTC cannot be determined. Therefore, the flow of Bitcoins can only be estimated based on the values of the inputs and outputs as shown in Table I.

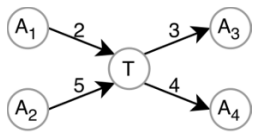


Fig. 2. Bitcoin transaction value assignment

Therefore, we estimate the flow of actual Bitcoins between two addresses using the following formula:

$$Est(I_i, O_j) = O_j * \frac{I_i}{\sum_k I_k}$$

TABLE I. BITCOINFLOW

Transaction	Formula	Estimated BTC
A1 → A3	3 * (2/7)	0.857
A2 → A3	3 * (5/7)	2.143
A1 → A3	4 * (2/7)	1.143
A2 →A4	4 * (5/7)	2.857

IV. ANALYSIS

A. Dataset

For our analysis, we took a dump of the Bitcoin blockchain which included all transactions from the first block on 3rd of January 2009 until block 430.000 on 15th of September 2016. The analysis was carried out month-wise and considers transaction data until 31st of August 2016. Table II shows the number of addresses, blocks, and transactions included in this dataset.

TABLE II. DATASET STATISTICS

Total number of addresses	176.412.948
Total number of blocks	430.000
Total number of transactions	156.365.848

Bitcoin users are encouraged to use each address only once, which means that ideally each Bitcoin address is involved in at most two - one receiving and one spending - transactions. The difference in the total number of addresses and number of transactions can be explained by *(i)* not all users following this recommendation and *(ii)* addresses that serve as an input for multiple transactions. Addresses are often reused by vendors and organizations that receive Bitcoin donations and refrain from anonymity on purpose by advertising their address publicly on the Web.

The numbers in Table II are also an indicator of the adoption of virtual currencies such as Bitcoin (BTC). While the number of organizations (e.g., Internet Archive[2]) and vendors who accept Bitcoin is still relatively low, the overall transaction volume is steadily increasing. An increase in reuse of addresses over time could in fact indicate a wider adoption by common vendors. An overview of vendors accepting Bitcoin for payment is available online[3] and amounts to around 8,400 worldwide (also including Bitcoin ATM).

B. Structural analysis

Given the growing number of transactions, we can expect that the structure of the Bitcoin address graph changes over time due to a growing number of participating users and organizations accepting Bitcoin for payment or donations.

For our Bitcoin address graph analysis and for our definition of in- and out-degree of single addresses, we represent addresses as vertices (nodes) and created a labelled directed edge for each transaction T that involved two addresses A_i and A_o as in- and output; that is, there may be multiple edges labelled with the same transaction T, in case multiple in- and outputs are involved. We associated each transaction T with several additional attributes, e.g. the transaction id, in which block the transaction occurred and the minimum, average and maximum transaction value transferred between these addresses.

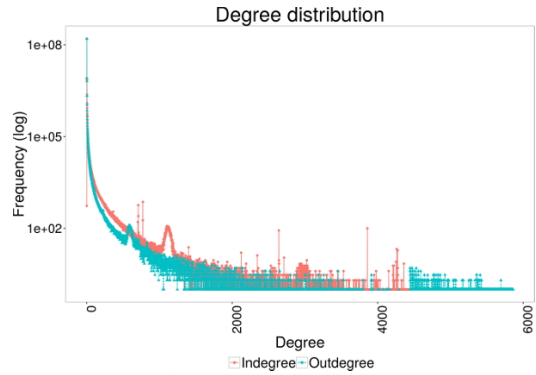


Fig. 3. Degree distribution

The degree of a vertex is defined as the number of incoming and outgoing edges. Based on the design and

[2] https://archive.org/donate/bitcoin.php (28.10.2016)
[3] http://www.coinmap.org

anonymity of the Bitcoin system, it is expected that most addresses have a low degree. However, since a single transaction can contain an arbitrary number of input and output addresses, the in-degree and out-degree of address nodes varies, as shown in Figure 3.

Especially high- and low-degree address nodes are of interest. Prominent examples for donation addresses are the Internet Archive[4] with an in-degree of 1,759 and an out-degree of 105 or WikiLeaks[5] (24,469/125). An address uniting both having the highest in-degree (1,595,498) and out-degree (1,600,277) belongs to the online Bitcoin casino satoshiDICE[6] (*1dice8EMZmqKvrGE4Qc9bUFf9PX3xaYDp)*.

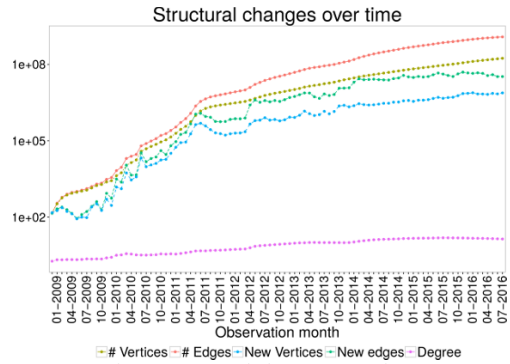


Fig. 4. Structural changes over time

Next, we analyzed the evolution of Bitcoin addresses over time. Figure 4 shows the cumulative number of distinct nodes (addresses) and edges in the address graph as well as the number of added nodes and edges and average node degree per month. We can observe that the number of used addresses is increasing, this can be interpreted in two ways: on one hand, it shows that the users use new addresses for each transaction, but this does not mean that each new address also leads to a new user. A single user can have lots of transactions within an observation period with many addresses. On the other hand, an increased usage of the virtual currency Bitcoin can be derived. The number of edges (transactions) is increasing in the same way. The degree remains almost steady over the course of time.

C. Real-world actors in the Bitcoin ecosystem

Transactions are anonymous by design and do not reveal the identities of real-world actors who can be individuals, exchanges, payment providers, or any other type of service in the Bitcoin ecosystem. However, as shown by previous research [3,8], it is possible to combine addresses into clusters (wallets), which are likely to be controlled by a certain real-world actor.

A well-known clustering heuristic works under the assumption that multiple addresses, used as input of a single transaction, must be controlled by the same real-world actor. This assumption holds if transactions are not executed through mixing services, which obfuscate the transaction by breaking the connection between a Bitcoin address sending coins and the addresse(s) they are sent to.

Other heuristics are based on the observation that *change*, is transferred back to the user when the sum of inputs is greater than the sum of outputs. Thus, one of the output addresses of a single transaction T often belongs to the same user or real-world actor; in a typical transaction, this address is even identical to one of the input addresses.

As soon as an address cluster (so called *entities*) is identified, a single address within that cluster carries an explicit tag with contextually relevant information, it is possible to implicitly assign that tag to all other addresses in the cluster and to possibly identify the real-world actor owning that address cluster, which often corresponds to a *wallet*. Therefore, we can group addresses into three different categories:

Unknown addresses: no tag has been assigned to an address and no contextual information is available publicly. From the point of view of the Bitcoin design, this is the desired ideal situation in terms of anonymity and address usage. Unknown addresses have not been used as input with other known addresses.

Explicitly known addresses: additional contextual information can be assigned in the form of a tag. Such tags can be extracted by crawling the Web or gathering data from external information sources such as blockchain.info[7], walletexplorer[8], social media platforms or the Darknet.

Implicitly known addresses: appear in a cluster with at least one other explicitly known address, from which tags can implicitly be derived. In the case shown in Figure 2, it is assumed that the addresses A_1 and A_2 are controlled by the same user and in addition, these addresses appear in a cluster with explicitly known addresses.

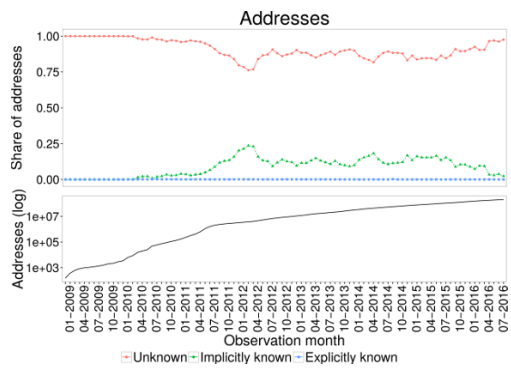


Fig. 5. Share of addresses by category.

Figure 5 shows that the fraction of explicitly known addresses is low throughout the entire Bitcoin's history. However, the fraction of implicitly known addresses starts

[4] https://archive.org/donate/bitcoin.php
[5] https://shop.wikileaks.org/donate
[6] https://www.satoshidice.com/
[7] http://www.blockchain.info
[8] http://www.walletexplorer.com

growing in 2010, reaches its maximum in 2012, remains roughly constant until 2015 and starts decreasing in 2016.

We assume that the decrease towards June 2016 is caused (i) by missing contextual information (tags) for newly generated addresses, (ii) the increasing awareness of end users that reuse of Bitcoin addresses decreases anonymity, and (iii) the increasing usage of Bitcoin mixing and tumbler services.

In general, the need for anonymity depends on the user group and the purpose for which Bitcoin is used. Organizations financing themselves with donations are well advised to publish their Bitcoin address on their homepage or social media to collect donations. They even generate so-called vanity addresses for that purpose, which are personalized addresses which often contain the organizations name in it (e.g. the addresses of organizations or gaming sites). On the other side of the spectrum are organizations conducting illegitimate business transactions such as collecting ransom from cybercrime activities.

D. Transaction behavior and exchange rates

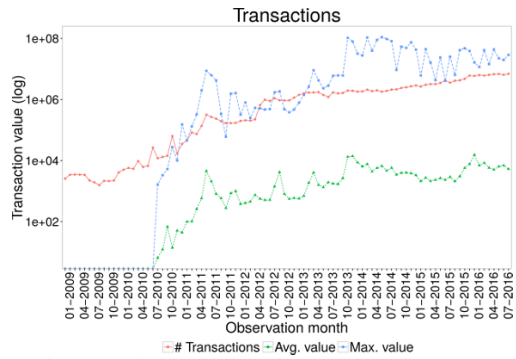


Fig. 6. Number and value of transactions (in USD)

Figure 6 shows the total number of transactions carried out in each month as well as the average and the maximum value of the transactions in USD with the exchange rate at the time of the transaction. Missing exchange rate data before 2010 caused the sudden increase in the value of the transactions.

The number of transactions is continuously growing during the first four years and has remained steady since then. An explanation could be that a virtual currency like Bitcoin was new and therefore an increasing number of people tried it out. After a few years, the interest in Bitcoin seemed to have flattened, but the market remained constant and the regular transactions remained. The peaks on the average transaction value in USD go hand in hand with the changes in the exchange rate. The fluctuating maximum real value can also be explained by the changes in the exchange rate, which will be explored in the following subsection.

The average exchange rate BTC/USD over time from January 2009 until December 2016 is shown in Figure 7 together with the minimum and the maximum per month. It is clearly visible, that the exchange rate remains steady for the first two years and afterwards shows volatile behavior. However, three months are particularly striking: April 2013, December 2013 and May 2016, where the difference between the minimum and the maximum prices for BTC is very high, whereas they are in a certain range for the rest of the months.

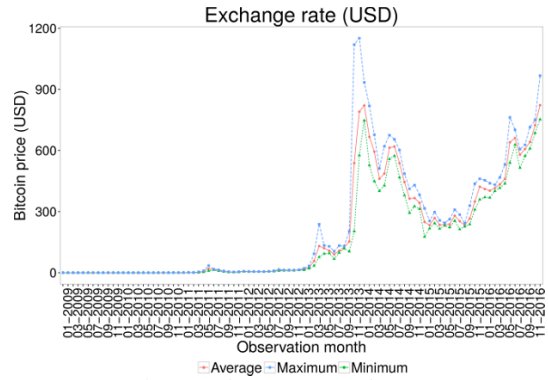


Fig. 7. Exchange rate BTC/USD

Real-world events influence the popularity of virtual currencies. The first peak in the exchange rate is visible in April 2013 shortly after the Cypriote government announced the bailout of Cyprus' banks causing many people to save their money by switching to Bitcoin[9]. Additionally, the seizure of Mt. Gox, a large Bitcoin exchange, had an impact on the exchange rate[10,11]. The highest price of around 1,250 USD was reached in late 2013, which marks the last noticeable rise in the number of transactions. The increase in value also caused an increase in the number of transactions. Investors strive to increase their wealth and a strongly increasing exchange rate of a currency is very tempting. The last clearly striking changes in the exchange rate occurred in July 2016. The military coup in Turkey caused a run to the Turkish banks as people wanted to know that their money was in a safe place[12]. Political events, which are likely to cause fear among the population, might also be reflected in the virtual currency markets.[13]

E. Activity time

Next, we examine the possible monetary functions (store of value, exchange of value) of Bitcoin by investigating how long users keep their Bitcoins and whether this currency is used as an alternative to long-term savings accounts.

TABLE III. BITCOINFLOW

Metric	Address based	Entity based
Avg. used in transactions	Incoming: 2.25 Outgoing: 1.75	Incoming: 10.5 Outgoing: 3.7
Avg. activity time (days)	12	15
Median activity time (days)	< 1	< 1

[9] http://money.cnn.com/2013/03/28/investing/bitcoin-cyprus/
[10] http://bit.ly/2nrTLGm
[11] http://bit.ly/2mN6J2O
[12] https://news.bitcoin.com/turkish-bitcoin-military-coup/
[13] http://bit.ly/2nmUdbU

Table III shows the active time period of addresses and entities. This is defined as the period between the first and last transaction a single address or an address within a cluster was involved in. The average activity time of an entity with all accounted addresses is 15 days (median: < 1 day), which strengthens the hypothesis that Bitcoin is not used as a replacement for saving accounts but rather as global payment system. This is in-line with our previous observation in Figure 3, showing that the degree for most addresses is rather small.

The number of entities with a positive balance on at least one of their associated addresses amounts to 368,739 with total unspent 1,565,294 BTC on the 15th September 2016. The mean balance is 4.25 BTC (median: 0.000795 BTC).

V. Conclusion

In this paper, we analyzed the Bitcoin address graph from several perspectives based on the publicly available ledger containing all transactions from the beginning of Bitcoin in January 2009 until 31st of August 2016.

We described the procedure we applied to construct the address graph; presented a possible strategy for estimating currency flows between addresses; and described the heuristics we applied for combining addresses in clusters of addresses, which can then be assigned to real-world actors.

Our structural analysis shows a highly-skewed degree distribution, which implies that the Bitcoin address graph comprises a small number of outliers with high in- and/or out-degree. Manual inspection of those addresses revealed that they are often used by (non-profit) organizations to receive donations or by online gambling Websites. Our analysis also shows that the address graph is expanding rapidly over time as new addresses and transactions are added to the blockchain. However, the average node degree remains stable over time.

Investigation of real-world actors showed that address clustering using well-known heuristics can increase the number of implicitly known addresses in the entire graph. However, most Bitcoin addresses remain anonymous.

Furthermore, our analysis illustrates the growing transaction volume and the effects of real-world events on the Bitcoin exchange rate. It also shows that real-world actors use Bitcoin more often for transferring value than for storing value. This indicates that Bitcoin is not used as an alternative to savings accounts, probably due to the above-mentioned volatility and instability of the currency.

A clear limitation of our work lies in the selection of tags, which affects the fraction of implicitly and explicitly known addresses. We expected that a more comprehensive tag dataset extracted from various sources would increase the fraction of identifiable addresses but not de-anonymize most addresses.

A possible direction for future work lies in the investigation of effects of external political or economic events (e.g., “Brexit”) on virtual currencies and the prediction of possible micro- and macroscopic reactions within or across virtual currency ecosystems. Furthermore, it would be interesting to extend the analytics methods presented in this paper to other crypto-currencies such as Monero or ZCash.

The dataset for our analysis is a cleansed graph representation of the blockchain and is available to other researchers on request.

VI. Acknowledgement

This work was funded by the Austrian research funding association (FFG) under the scope of the ICT of the Future program (contract # 849906).

References

[1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system.” 2008.

[2] D. Ron and A. Shamir, “Quantitative analysis of the full bitcoin transaction graph,” in *International Conference on Financial Cryptography and Data Security*. Springer, 2013, pp. 6–24.

[3] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, “A fistful of bitcoins: characterizing payments among men with no names,” in *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 2013, pp. 127–140.

[4] A. Biryukov, D. Khovratovich, and I. Pustogarov, “Deanonymisation of clients in bitcoin p2p network,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 15–29.

[5] J. V. Monaco, “Identifying bitcoin users by transaction behavior,” in *SPIE Defense + Security*. International Society for Optics and Photonics, 2015, pp. 945 704–945 704.

[6] M. Fleder, M. Kester and S. Pillai, “Bitcoin Transaction Graph Analysis”, CoRR, vol. abs/1502.01657, 2015. [Online] Available: http://arxiv.org/abs/1502.01657.

[7] M. Möser, R. Böhme, and D. Breuker, “An inquiry into money laun- È dering tools in the bitcoin ecosystem,” in *eCrime Researchers Summit (eCRS)*, 2013. IEEE, 2013, pp. 1–14.

[8] F. Reid and M. Harrigan, “An analysis of anonymity in the bitcoin system,” in *Security and privacy in social networks.* Springer, 2013, pp. 197–223.

[9] B. Holtz, J. Fortuna and J. Neff, “Evolutionary structural analysis of the bitcoin network”, 2013.

[10] A. Miller, J. Litton, A. Pachulski, N. Gupta, D. Levin, N. Spring and B. Bhattacharjee, “Discovering Bitcoin’s Public Topology and Influential Nodes”, 2015.

[11] M. Harrigan and C. Fretter, “The unreasonable effectiveness of address clustering”, CoRR, vol. abs/1605.06369, 2016. [Online] Available: http://arxiv.org/abs/1605.06369.

[12] D. Kondor, M. Posfai, I. Csabai, and G. Vattay, “Do the rich get richer? An empirical analysis of the bitcoin transaction network,” *PloS one*, vol. 9, no. 2, p. e86197, 2014.

[13] M. Ober, S. Katzenbeisser, and K. Hamacher, “Structure and anonymity of the bitcoin transaction graph,” *Future internet*, vol. 5, no. 2, pp. 237–250, 2013.

[14] B. Haslhofer, R. Karl, and E. Filtz, “O bitcoin where art thou? insight into large-scale transaction graphs,” in Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - SEMANTiCS2016 and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS’16) co-located with the 12th International Conference on Semantic Systems (SEMANTiCS 2016), Leipzig, Germany, September 12-15, 2016., ser. CEUR Workshop Proceedings, M. Martin, M. Cuquet, and E. Folmer, Eds., vol. 1695. CEUR-WS.org, 2016. [Online]. Available: http://ceur-ws.org/Vol-1695/paper20.pdf

[15] M. Spagnuolo, F. Maggi, and S. Zanero, “Bitiodine: Extracting intelligence from the bitcoin network,” in *International Conference on Financial Cryptography and Data Security*. Springer, 2014, pp. 457–468.

Data Analytics in Industrial Application Scenarios

A Reference Architecture for Quality Improvement in Steel Production

David Arnu, Edwin Yaqub
RapidMiner GmbH
Dortmund, Germany

Claudio Mocci, Valentina Colla
TeCIP Institute
Scuola Superiore Sant'Anna
Pisa, Italy

Marcus Neuer
VDEH-Betriebsforschungsinstitut GmbH
Düsseldorf, Germany

Gabriel Fricout, Xavier Renard
ArcelorMittal Maizières Research SA
Maizières-Les-Metz, France

Christophe Mozzati
PREDICT SAS
Vandoeuvre lès Nancy, France

Patrick Gallinari
LIP6
Universite Pierre et Marie Curie
Paris, France

***Abstract*—There is a global increase in demand for steel, but steel manufacturing is a highly sophisticated and costly process where good quality is hard to achieve. Improving the quality remains a major challenge faced by the steel industry. The EU project PRESED (Predictive Sensor Data mining for Product Quality Improvement) addresses this challenge by focusing on widespread recurring problems. The variety and veracity of data, as well as the change in properties of the observed material complicates the interpretation of data. In this paper, we present the reference architecture of PRESED, which is being purpose-built to address the vital concerns of managing and operationalizing the data. The architecture leverages big and smart data concepts with data mining algorithms. Data preprocessing and predictive analytics tasks are supported by means of a malleable data model. The approach allows the users to design processes and evaluate multiple algorithms pertinent to the problem at hand. The concept is to store and harness the complete production data instead of relying on aggregated values. Early results on data modeling show that fine grained preprocessing of time series data through feature extraction and predictions provide superior insights than traditionally used aggregation statistics.**

Keywords—process optimization; steel manufacturing; data mining; time series; nosql

I. INTRODUCTION

European steel industry is facing extremely challenging global markets with very strong international competition. It is a fact that costs for manpower, raw materials and energy are significantly lower in other regions of the world like China, India or Brazil. It is therefore indispensable for Europe to exploit its current production facilities in a much smarter way to keep its competitive edge. Therefore, novel methods to improve the manufacturing of steel products are essential for market success.

Steel production is a complex process comprising different steps. Each of these steps has to be very precisely mastered in terms of process conditions (temperature, casting speed, cooling flow rate, etc.), as slight deviations can lead to the occurrence of defects on the product. Mostly, physical understanding enables to target the origin of the problem. But in situations where a clear understanding is not available, data mining is a key technology for addressing and clarifying the origin of the problem. Due to the complexity of the underlying processes and the data, in particular material tracking over the complete production cycle, sophisticated tools have to be developed.

In this work, we present a conceptual architecture for storing and processing the massive amount of sensor data that are created during steel manufacturing. We aim to build a reference model that is able to take advantage of new data storage methods (e.g., NoSQL) and allows the usage of predictive analytic methods on the stored data.

The layout of this paper is as follows. In Section II, we outline the state of the art for quality management and predictive analytics. An overview of current big data technologies and their relevance to the PRESED project is also presented. In Section III, we give an overview of the manufacturing process in steel plants. Section IV presents the PRESED architecture defined by the PRESED project, its data model, a summary of applicable algorithms and the embedded ontology design. We conclude the paper in Section V and highlight future directions.

This project is funded by the European Commission Research Program of the Research Fund for Coal and Steel Technical Group: TGS9/ Grant number RFSR-CT-2014-00031

P. Haber et al. (Hrsg.), *Data Science – Analytics and Applications*, http://doi.org/10.1007/978-3-658-19287-7_12

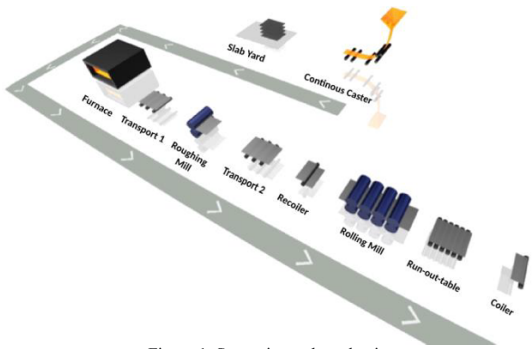


Figure 1: Stages in steel production

II. Related Work

In the past, data mining helped to perform quality surveillance for industrial processes [1] and was used along the production route of steel [2], often with a focus on surface quality [3] or root cause analysis [4]. These tools evaluated several relevant but static quantities to find the root causes of certain production issues. These approaches were limited to a small subset of information about the product, simply because the amount of available data was too large and several aggregations were applied before the actual mining took place. Another work that applied data mining to the steel industry is presented in [5], where the focus is on data processing methods.

The here presented architecture approaches data mining on sensorial time series data. This emphasis on uncompressed data differs significantly from traditional methods as it requires a thorough analysis of the time series. Storing the data in a relatively unprocessed state and applying data analytical processes is a common paradigm for big data architectures, e.g., the Lambda Architecture by Nathan Marz [10], which targets key-value paired data.

In contrast to the referred works, the novelty of our work lies in *product-orientation*. To efficiently store information belonging to a product, each product – in the following called a metal unit - has a virtual representation in a NoSQL database. The metal unit can be a heat, a slab or a coil, depending on the physical state of the product. Adopting such a view is new to steel industry and was not used in previous works. We decided to use a NoSQL database for storing the metal unit, as it does not depend on a fixed uniform data schema, but other storage models are also possible.

Also, a series of dedicated algorithms have been developed for the PRESED project, that significantly extend the common state of the art in the field of predictive data mining. Among them are outlier detection, Self-Organizing Maps and pattern discovery methods that were trained with data from the use cases. Synergies were found between the product-oriented data concept and the machine learning algorithms, as the new data concept allows an easy and flexible storage of multiple labels per product.

III. Domain of Steel Manufacturing

Manufacturing steel is a time-sensitive and multi-step process as illustrated in Fig. 1. A typical process flow to produce steel for flat products (sheet) involves the following steps:

- **Iron making**, where liquid iron is produced either from iron ore (blast furnace) or from scraps (electric arc furnace)
- **Steel making**, where the chemistry of the liquid metal is progressively adjusted through several reactors in order to reach the expected target.
- **Continuous casting**, where the liquid steel solidifies into one slab – a piece of solid steel.
- The slab is then re-heated and **hot-rolled** to have a first length adjustment and produce what is called a "coil". For flat products, the length of a slab is typically 20 meters, whereas the length of a coil can reach several hundreds of meters.
- Another step is **cold-rolling** for further elongation of the coil. Depending on the target thickness, the final length can reach several kilometers.
- **Annealing and galvanizing**: a thermal cycle is applied on the steel to adjust its mechanical properties before a zinc coating is applied against corrosion.

A. Use cases

For developing the PRESED architecture and validating our approach, the following use cases are considered:

(1) Predict Sliver Occurrence: Slivers are one of the major sources of quality loss on the steel surface. The defect occurs at the continuous casting step, when slight amount of slag (various liquid oxides remaining on the top of the continuous casting machine) are entrapped at the liquid steel surface during the solidification process. The occurrence of the defect is strongly impacted by the liquid steel flow in the continuous casting machine,

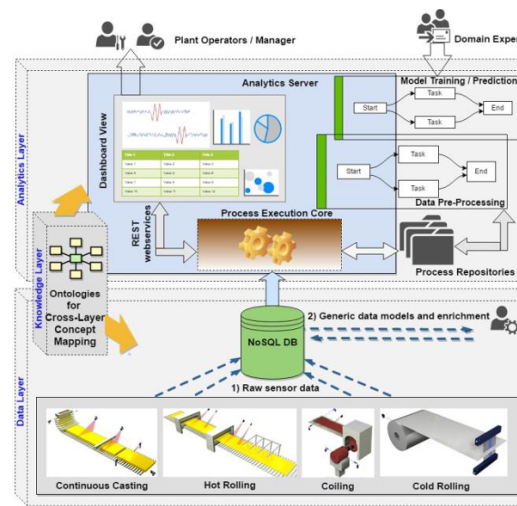


Figure 2: Architecture diagram depicting key concepts

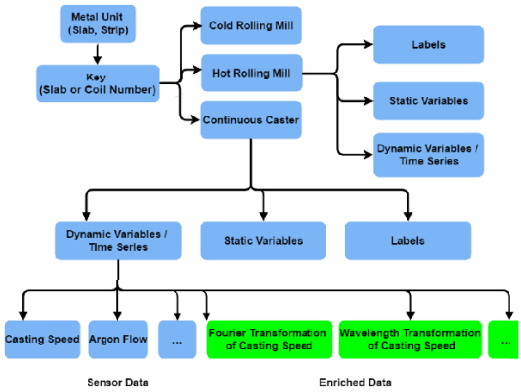


Figure 3: The generic data model for the metal unit.

which is itself dependent in a very complex way from all process conditions (temperature, speed, continuous casting actuators, etc.). This is a major concern, because the defect is usually detected at the final step of the process, leading to very high re-allocation costs.

(2) Scattering of Mechanical Properties (MP): MP are one of the most important product criteria for the customer. If the specifications are not reached, a lot of issues can be experienced by the steel customer, in particular for stamping operations. For certain steel grades, these mechanical properties are very sensitive to slight variations of temperature during the last annealing cycle. The thermal behavior of the steel in the last furnace is itself highly dependent on process parameters ranging from continuous casting to the galvanizing lines. A bad mastering of these parameters leads to scattering in mechanical properties, but the study of the phenomenon is very complex since data from all along the production chain has to be considered.

(3) Production Process Chain: This use case aims at controlling the (non-Sliver related) surface defects and improving the inner quality of steel product by considering all phases of the production chain between electric furnace and continuous casting.

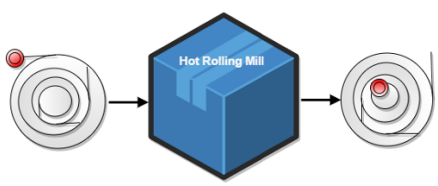


Figure 4: Effect of coiling and de-coiling on length and observed position points

IV. ARCHITECTURE

The proposed architecture addresses various concerns for data storage, data mining, visualization and concept formalism. It is quite evident from previous sections that a singular software framework can hardly address the diverse requirements for our use cases. Instead, we emphasize on an extensible approach which can integrate the following technical requirements:

- A generic malleable data model for preprocessing and enriching the raw sensor data.
- Application of various algorithms to the problem at hand e.g., data transformation, feature extraction, outlier detection or classification.
- Designing and executing data mining processes that encompass the above concerns in a visual and concomitant manner to support reuse and sharing.

The PRESED architecture serves the needs of three major stakeholders: **1)** The *Data Engineers*: They model the raw data into a generic model (Section A-1). This is a preparatory activity which may be done infrequently. **2)** The *Domain Experts*: They author data mining processes on the now unified data model. **3)** The *Plant Operators* and *Managers*: These actors execute the previously created processes to seek advanced insights on product quality and potential defects. The results are presented visually to assist human interpretation and enable timely corrective measures.

At the high level, the design splits these concerns into three planes: a *data layer*, an *analytics layer* and a *knowledge layer* as shown in Fig. 2. The salient features of these layers along with the challenges faced are presented below in detail.

A. Data Layer

The data layer deals with the set of sensor readings from various stages of production e.g., continuous casting, hot or cold rolling, coiling and uncoiling of the metal unit. This “raw” time series data is encapsulated in an object representing the metal unit and stored in a NoSQL database. Although the volume of data collected so far is not as extensive as in other big data projects, it is expected to grow in future. We selected MongoDB because of the simplicity it offers to store unstructured and sparse data as objects that can be efficiently retrieved. It further allows the user to update an object or entire collection of objects with new attributes as or when the need for data enrichment arises. This is well suited to industrial settings where sensor values become dynamically available and where readings corresponding to a certain stage of the production process need be inspected. In so, the variety and veracity aspects of the big data paradigm play a strong role in this work.

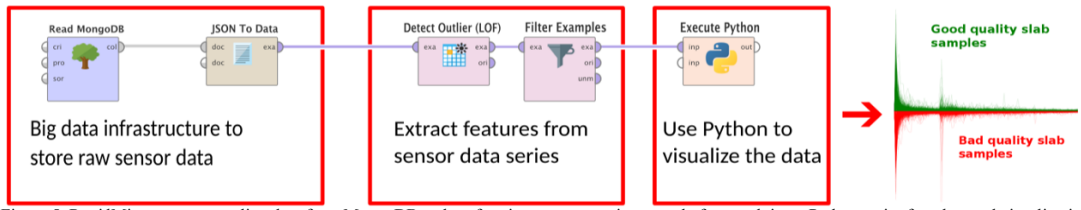


Figure 5: RapidMiner process reading data from MongoDB and performing pre-processing steps before applying a Python script for advanced visualization (left). Fourier transformation of the mould level. Overlay of 1000 slabs, divided into good (green) and bad (red) label classes (right)

1) Generic Data Model

The structure of the metal unit (a coil or slab of steel or a heat) is composed as a generic data model as shown in Fig. 3. The unit is identified by its identifier and has aggregated nodes that hold data from its production process (casting, hot or cold rolling, etc.) or data derived through enrichment (depicted as green nodes). In general, data for each of these categories exist either as static variables which hold fixed univariate values or as dynamic variables which hold time series data. Additionally, information about the unit such as inspection data, cuts, parent/child units can also be linked through identifiers.

The datasets being used are real data from steel plants in France and Italy. For the first two use cases, there exists data for 2261 coils, of which 1205 were labeled (or classified) as good and 1056 of bad quality. The data also includes 5000 slabs, with casting speed, actuator position, mould level, argon gas flow and pressure values - collected during the continuous casting phase. Further, a set of 70 static and 125 dynamic variables are considered for data preparation. The data for the third use case treats a heat as a metal unit. It encompasses the process as steel goes from the electric furnace to the continuous casting phase. A heat in this context refers to the molten metal. The data include 10,000 heats collected over a 2-year period.

2) Achieving Smart Data through Enrichment

Before applying a learning algorithm, the data needs to be pre-processed. Concerning time series data, the volume of data is not the only key factor. To deliver qualitative reliable information, additional enrichment techniques are required. The first steps are sanity checks such as the usage of consistent units, treating missing values and normalization of data ranges. The latter also serves to anonymize the possibly sensitive data when sharing them among project partners. The metal unit may change in length due to the milling or cutting of the product. Further, as a result of coiling and de-coiling, the head of the coil transforms to the tail point and vice-versa as shown in Fig. 4.

For situations like these, where the geometry inside the metal unit changes, or wherever different sampling intervals are used for sensorial data acquisition at different stages, rescaling is done by interpolating all data with inferior sampling points onto the discretization of highest resolution. This preserves the information for the higher resolved data, which would be lost in the case down-sampling.

3) Systematic Process Design

The next step is to systematically compose these repetitive operations as an executable data mining process. This activity is supported through the open-source and free-to-use RapidMiner [8] tool that allows to graphically create data mining processes. Processes resemble a pipeline composed of tasks that are wired together through drag and drop mechanism. Tasks may perform ETL (Extract, Transform and Load) or machine learning operations by invoking different algorithms and evaluating their performance.

For instance, Fig. 5 illustrates how data enrichment is applied through transformation functions on the raw sensor values. First, the process reads data from the MongoDB instance and converts the JSON format into an in-memory representation. Next, feature extraction is performed by first applying an outlier detection algorithm and then filtering the data set based on the outlier score. Finally, a Python script is invoked to generate a visualization which shows the Fourier transformation of the impurity (mould) level, where 1000 slabs are shown - classified as having good or bad quality.

Such transformation processes help to detect or highlight certain properties of signals e.g., calculating the derivation or applying a signal space transformation. Because of the malleable data schema, it is possible to pre-calculate the transformation and append the results to an existing metal unit. Thus, even the results of computationally intensive transformations are available for reuse. Complex processes for predictive analytics are designed and applied in the same way.

B. Analytics Layer

The analytics layer caters for the management and execution of data mining processes. This layer centers around the Analytics Server which provides: **1)** A repository for storing RapidMiner processes. This eases collaboration by providing shared access. **2)** An execution core to execute processes upon demand. **3)** A dashboard view that allows to browse the data and display execution results as charts (using Python and/or JavaScript libraries). These visualizations can be customized through query parameters because the Analytics Server exposes processes as REST-full web services. The latter paves the way for interoperability with legacy systems and operationalizing on predictions.

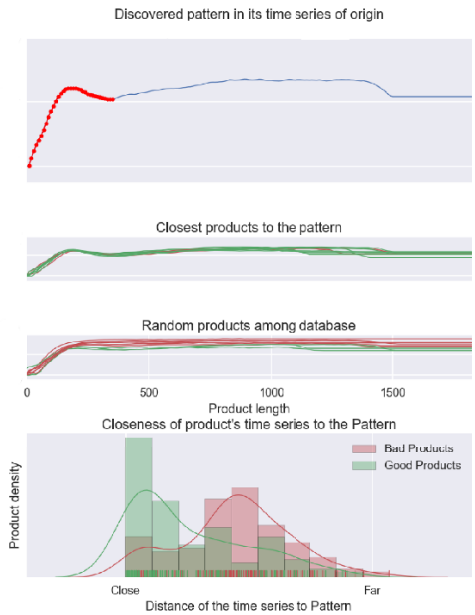


Figure 6: An example of temporal pattern discovery using the shapelet algorithm.

1) Feature Extraction

When dealing with multivariate time series, it is essential to extract the most relevant features. The information contained in the dynamic aspects of sensor data (temporal evolution of the measurements) faces specific issues, which are:

- The relevant temporal information is typically encoded in many complex ways such as segments, spikes, periodicity, drifts or often a combination of these. The relevant information can be the whole signal or only a sub-sequence.
- In an industrial context, the time series are usually multivariate: dozens of sensors measure process parameters at the same time or at the same positions of the product.
- Time series are particularly prone to noise: the typical measurement noise and process variability are duplicated by the successive observations. Time series also suffer from the time-axis noise named time warping: the same phenomenon can occur at several speeds and also suffers from local misalignment.
- When comparing time series, the high dimensionality is a prominent issue as the number of measurements is usually very large – this is often referred in mathematics as the "curse of dimensionality" [9].

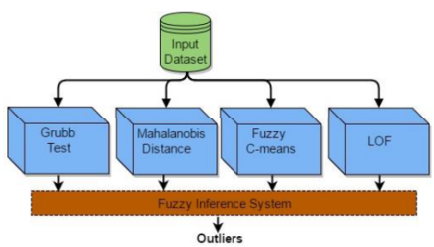


Figure 7: Outlier detection algorithm

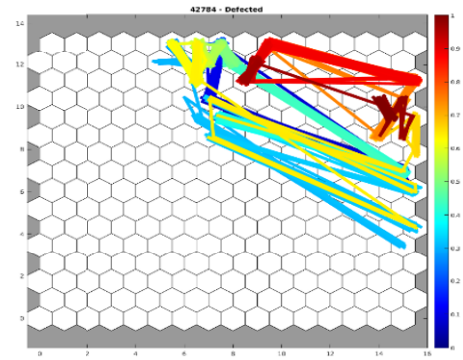


Figure 8: SOM hits plot with trajectory

Those problems are not restricted to the use cases discussed here and there are existing solutions, e.g., low-pass filters, dynamic time warping, dimensionality reduction algorithms. These solutions are incorporated into the data enrichment process.

In interaction with process experts, a supervised temporal pattern discovery approach based on the shapelet concept [12] was developed (Fig. 6). The objective was to discover localized discriminant sub-sequences in the time series. The search is driven by the product quality information. In previous work [11], we addressed the scalability issue of the discovery process to apply the method on large datasets. Furthermore, a generalization of the shapelet concept was developed to efficiently discover diverse shapelets with respect to the whole context.

2) Outlier Detection

Detecting deviant observations can help to identify possible problems in the production during an early stage. The main idea of the proposed outlier detection algorithm ensemble is to combine different approaches (density, clustering, distribution and distance) by means of a fuzzy inference system (Fig. 7). Due to the automatic computation of all the parameters needed to execute the elaboration of the algorithm, no a-priori knowledge is required. The fuzzy inference system is further described in [6].

3) Predictive Models

Another aspect addressed in PRESED regards the predictive modeling for process control. For instance, the third use case applies an unsupervised learning approach to detect anomalous behavior of one or more continuous casting process variables.

This has been achieved by means of a SOM (Self Organizing Map) [7].

The cause of defected heats might be the sudden changes in values of specific process variables. This phenomenon can be visualized by looking at a customized SOM hits plot, by adding trajectories of the activated neurons to the net (Fig. 8).

C. Knowledge Layer

The objective of the Knowledge Layer is to facilitate an exchange between process and data mining experts. To do so, concepts relative to steel manufacturing are modeled together with concepts relative to data processing, as seen in Fig. 9. These are formalized in an ontology (using the OWL[1] language) that contains concepts, instances and rules. The list of concepts combines a list of common defects and physical concepts, characterizing a metal product (such as density or crystal structure). Rules (described using the SWRL[2] language) link a defect to its effect in the final product.

To exploit this ontology, a web portal is developed based on the KASEM [13] software. The portal allows to query the knowledge stored inside the ontology. The objective is to enable process experts to find a suitable algorithm for a given problem scenario. Using theoretical knowledge combined with the results from experiences of previous use cases, the software can advise the best data processing algorithm(s) for this specific problem. The high-level model in the ontology also allows this application to be generalized in a fleet-wide approach. It is thus possible to access knowledge gained in another plant for similar situations and have more precise suggestions at hand [14]. For example, a query can show the used data enrichment processes or the applied algorithms and their parameters for a new, but similar metallurgical problem.

V. Conclusion and Future Work

Improving the quality of steel production processes has been a long-term goal for the industry. The PRESED architecture addresses these goals by leveraging big and smart data technologies with data processing and mining techniques. The on-going progress on use cases is expected to lead to novel results which may serve as a niche for the steel industry in improving the product quality as well as optimizing the whole production processes. Future extensions and improvements are planned especially regarding operationalizing of results in real plants. We also plan to curate the process web services to form a unified API for broader adoption of PRESED architecture. Finally, a link between KASEM and RapidMiner will be investigated to use the knowledge stored in the ontology for context-driven algorithm instantiations.

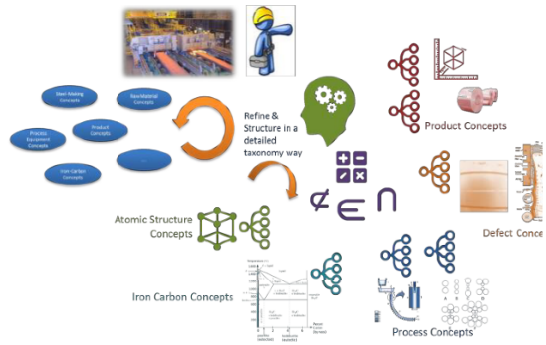


Figure 9: Modeling of steel and data processing concepts inside an ontology

References

[1] J. Ordieres-Mer□, F. Alba-El□as, A. Gonz□lez-Marcos, M. Castej□n-Limas, F. J. Pis□n-Ascac□bar, Data mining and simulation processes as useful tools for industrial processes, Proc. 5th WSEAS Int. Conf. Simulation, modelling and optimization, pp249-255, 2005

[2] J. Ordieres-Meré, M. Castejón-Limas, Data Mining Applications in Steel Industry, IGI-Global, 2006

[3] S. Mehran Sharafi, H. Reza Esmaeily, Applying data mining methods to predict defects on steel surface, J. Th. Appl. Inf. Technology, Oct. 2010

[4] H. Peters, A. Ebel, J. Hackmann, M. Pander, Industrial data mining in steel industry, StahlEisen, Vo. 132 (2), December 2012

[5] J. Deuse, B. Konrad, D. Lieber, K. Morik, M. Stolpe, Challenges for Data Mining on Sensor Data of Interlinked Processes, SFB 876, 2011

[6] Cateni Silvia, Valentina Colla, and Gianluca Nastasi. "A multivariate fuzzy system applied for outliers detection." *Journal of Intelligent & Fuzzy Systems* 24.4 (2013): 889-903.

[7] Gianluca Nastasi, Claudio Mocci, Valentina Colla, Frenk Van Den Berg, Willem Beugeling. SOM-based analysis to relate non-uniformities in magnetic measurements to Hot Strip Mill process conditions. Proceeding of the 19th World Conference of Non-Destructive Testing (WCNDT) 13-17 June 2016, Munich, Germany

[8] Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., & Euler, T. (2006, August). Yale: Rapid prototyping for complex data mining tasks. In*Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 935-940). ACM.

[9] Hastie, T., Tibshirani, R., & Friedman, J. (2001). The elements of statistical learning. 2001. NY Springer.

[10] Marz, N., & Warren, J. (2015). *Big Data: Principles and best practices of scalable realtime data systems*. Manning Publications Co.

[11] X. Renard, M. Rifqi, G. Fricout, M. Detyniecki : "EAST representation: fast discovery of discriminant temporal patterns from time series", ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data, Riva Del Garda, Italy (2016)

[12] X. Renard, M. Rifqi, W. Erray, M. Detyniecki : "Random-shapelet: an algorithm for fast shapelet discovery", 2015 IEEE International Conference on Data Science and Advanced Analytics (IEEE DSAA©2015), Paris

[13] Monnin, M., L□ger, J. B., & Morel, D. KASEM: e-Maintenance SOA platform. In Proceedings of 24th International Congress on Condition Monitoring and Diagnostics Engineering Management, 29th May–1st June, Stavanger, Norway (2011)

[14] G. Medina-Oliva, F. Peysson, A. Voisin, M. Monnin, J-B Leger, Ships and marine diesel engines fleet-wide predictive diagnostic based on ontology, improvement feedback loop and continuous analytics, Proceedings of 26th International Congress of Condition Monitoring and Diagnostic, Engineering Management COMADEM, Helsinki, Finland, 2013

[1] Ontology Web Language

[2] Semantic Web Rule Language

Anomaly Detection and Structural Analysis in Industrial Production Environments

Martin Atzmueller
Tilburg University (TiCC),
Jheronimus Academy of Data
Science, University of Kassel (ITeG)

David Arnu
RapidMiner GmbH
Dortmund, Germany

Andreas Schmidt
University of Kassel (ITeG)
Kassel, Germany

***Abstract*—Detecting anomalous behavior can be of critical importance in an industrial application context. While modern production sites feature sophisticated alarm management systems, they mostly react to single events. Due to the large number and various types of data sources a unified approach for anomaly detection is not always feasible. One prominent type of data are log entries of alarm messages. They allow a higher level of abstraction compared to raw sensor readings. In an industrial production scenario, we utilize sequential alarm data for anomaly detection and analysis, based on first-order Markov chain models. We outline hypothesis-driven and description-oriented modeling options. Furthermore, we provide an interactive dashboard for exploring and visualization of the results.**

Keywords—anomaly detection; exceptional model mining; sequence mining; sequential patterns; industry 4.0

I. Introduction

In many industrial areas, production facilities have reached a high level of automation: sensor readings are constantly analyzed and may trigger various forms of alarms. Hence, knowledge about the respective processes is crucial, e.g., targeting the topological structure of a plant, sequences of operator notifications (alarms), and unexpected (critical) situations. Then, the analysis of (exceptional) sequential patterns is an important task for obtaining insights into the process and for modelling predictive applications. The research project *Early detection and decision support for critical situations in production environments* (short FEE) aims at detecting critical situations in production environments as early as possible and to support the facility operator in handling these situations, e.g., [14]. In abnormal situations, typically such a large number of notifications is generated, that it often cannot be physically assessed by the operator [2]. Therefore, appropriate abstractions and analytics methods are necessary to adapt from a reactive to a proactive behavior. The consortium of the FEE project consists of several partners also including application partners from the chemical industry. These partners provide the use cases for the project and background knowledge about the production process which is important for designing suitable analytical methods.

This paper presents the implementation of a comprehensive modelling approach for anomaly detection and analysis of observed "reference" sequential patterns, based on methods for modelling and comparing hypotheses on sequence data [16, 17]. Implemented as a new RapidMiner operator and embedded in an analytical process, we demonstrate its application.

II. Related Work

The investigation of sequential patterns and sequential trails are interesting and challenging tasks in data mining and network science, in particular in graph mining and social network analysis, e.g., [5, 9]. A general view on modeling and mining of ubiquitous and social multi-relational data is given in [5] focusing on social interaction networks. Here, dynamics and evolution of contact patterns [9, 23, 29], for example, and their underlying mechanisms, e.g., [33] are analyzed. However, the analysis in these contexts focuses on aggregated sequential data. Navigational patterns, as sequential (link) patterns in online systems, have been analyzed and modeled, e.g., in [35, 40]. In contrast to that, our approach focuses on the modeling and comparing sequential patterns (hypothesis) in a graph-based network representation.

In a previous work [16] we have presented the DASHTrails approach that incorporates probability distributions for deriving transitions utilizing HypTrails [39]. Based on that, the HypGraphs framework [17] provides a more general modeling approach. Using general weight-attributed network representations, we can infer transition matrices as graph interpretations; HypGraphs consequently also relies on Markov chain modeling [29, 35] and Bayesian inference [35, 41].

Sequential pattern analysis has also been performed in the context of alarm management systems, where sequences are represented by the order of alarm notifications. Folmer et al. [21] proposed an algorithm for discovering temporal alarm dependencies based on conditional probabilities in an adjustable time window. To reduce the number of alarms in alarm floods Abele et al. [2] performed root cause analysis with a Bayesian network approach and compared different methods for learning the network probabilities. Vogel-Heuser et al. [41] proposed a pattern-based algorithm for identifying causal dependencies in the alarm logs, which can be used to aggregate alarm information and therefore reduce the load of information for the operator. In contrast to those approaches, we provide a systematic approach for the analysis of sequential transition matrices and its comparison relative to a set of hypotheses. Thus, similar to evidence networks in the context of social networks, e.g., [32], we model transitions assuming a certain interpretation of the data towards a sequential representation. Then, we can identify important influence factors.

This work was funded by the BMBF project FEE under grant number 01IS14006

P. Haber et al. (Hrsg.), *Data Science – Analytics and Applications*, http://doi.org/10.1007/978-3-658-19287-7_13

Process Mining [1] aims at the discovery of business process related events in a sequence log. The assumption is that event logs contain fingerprints of a business process, which can be identified by sequence analysis. One task of process mining is conformance checking [34, 37] which has been introduced to check the matching of an existing business process model with the segmentation of the log entries. Compared to these approaches, we do not use any apriori knowledge about business processes to create our hypothesis. Also, our hypothesis does not necessarily need to conform to an existing business process.

One definition of an anomaly or outlier is, that "an outlier is an observation that differs so much from other observations as to arouse suspicion that it was generated by a different mechanism" [22]. Thus, we understand an anomaly as a real-world situation, that could be represented as one or more outliers. In the literature those two terms are often used interchangeably. Then, interesting, important or exceptional groups [36] can be identified. Classic approaches for anomaly detection provide a classification of anomalous and normal events. In the industrial context finding either significant changes in the multivariate sensor readings or managing hundreds of univariate scores for single sensors is also a challenge, e.g., see [30]. In contrast to approaches for anomaly detection that only provide a classification of anomalous and normal events, we can assess different anomaly hypotheses: Applying the proposed approach, we can then generate an anomaly indicator – as a potential kind of second opinion method for assessing the state of a production plant that can help for indicating explanations [15] and traces of unusual alarm sequences in the plant. Also, using the network representation, we can analyze anomalous episodes relative to structural (plant topology) as well as dynamic (alarm sequence) episodes.

III. Method

The detection and analysis of irregular or exceptional patterns, i.e., anomalies, in complex-structured heterogeneous data is a novel research area, e.g., for identifying new and/or emerging behavior, or for identifying detrimental or malicious activities. The former can be used for deriving new information and knowledge from the data, for identifying events in time or space, or for identifying interesting, important or exceptional groups. In this paper, we focus on a combined detection and analysis approach utilizing heterogeneous data. That is, we include semi-structured, as well as structured data for enhancing the analysis. Furthermore, we also outline a description-oriented technique that does not only allow the detection of the anomalous patterns, but also its description using a given set of features. The latter relates to the context of descriptive pattern mining. In particular, the concept of exceptional model mining, e.g., [8, 25, 27] suitably enables such description-oriented approaches, adapting methods for the detection of interesting subgroups (that is, subgroup discovery) with more advanced target concepts for identifying exceptional (anomalous) groups.

In our application context of industrial production plants in an Industry 4.0 context, cf., [20, 42], we based our anomaly detection system on the analysis of the plant topology and alarm logs as well as on the similarity based analysis of metric sensor readings. The combined approach will compare the insights of the two methods.

1) *Anomaly Analytics on Sequential Data*

For sequential data, we formulate the "reference behavior" by collecting episodes of normal situations, which is typically observed for long running processes. Episodes of alarm sequences (formulated as hypotheses) can be compared to the normal situations in order to detect deviations, i.e., abnormal episodes. We map these sequences to transitions between functional units of an industrial plant, applying the modeling approach described below. The results can also be used for diagnostics, by inspecting the transitions in detail.

Following HypGraphs [18] and DASHTrails [17], we model transition matrices given a probability distribution of certain states. The steps we need to perform, as shown in Fig. 1, are:

1) Modeling: Determine a transition model given the respective weighted network using a transition modeling function $\tau : \Omega \times \Omega \rightarrow R$. Transitions between sequential states $i,j \in \Omega$ are captured by the elements $m_{i,j}$ of the transition matrix M, i.e., $m_{i,j} = \tau(i,j)$. Then, we collect sequential transition matrices for the given network (data) and hypotheses.
2) Estimation: Apply HypTrails [39] on the given data transition matrix and the respective hypotheses, and return the resulting evidence.
3) Analysis: Present the results for semi-automatic introspection and analysis, e.g., by visualizing the network as a heatmap or characteristic sequence of nodes.

We can model (derived) transition matrices corresponding to the observed data, e.g., given frequencies of alarms on measurement points, as well as hypotheses on sequences of alarms. For data transition matrices, we need to map the transitions into derived counts in relation to the data; for hypotheses, we provide (normalized) transition probabilities.
In summary, we utilize Bayesian inference on a first-order Markov chain model. As an input, we provide a (data) matrix, containing the transitional information (frequencies) of transition between the respective states, according to the (observed) data. In addition, we utilize a set of hypotheses given by (row-normalized) stochastic matrices, modelling the given hypotheses. The estimation method outputs an evidence value, for each hypothesis, that can be used for ranking. Also, using

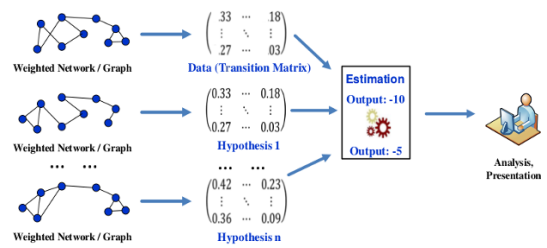


Figure 1: Overview on the HypGraphs modeling and analysis process [17].

Figure 2: RapidMiner Dashboard showing the HypGraphs transition scores (left) and the raw transition matrices (right), as transitions between different components of a plant visualized as a heatmap. It is possible to select a specific hypothesis for which the evidence scores are calculated and displayed.

the evidence values, we can compare the hypotheses in terms of their significance.

For modeling, we use the freely available RapidMiner [31] extension of HypGraphs[1], that calculates the evidence values for different believe weights k and compares them directly with the given hypothesis and a random transition as a lower bound. The evidence scores and transition matrices are displayed in an interactive dashboard, as shown in Figure 2.

As an extension to the hypothesis-based approach, we can furthermore include descriptive information, that is, features of the dataset for identifying patterns capturing anomalous behavior [11]. For that, we can consider the transition matrix as a graph, for which we can then we include node and/or edge labels. That is, the edges of the graph (modeling specific transitions between nodes) are labeled according to descriptive properties, e.g., capturing properties of the specific sequences the transitions were derived from. Then, using a specific set of labels we can select a set of edges, that is, all edges having the respective label set, inducing a subgraph which corresponds to a set of transitions having the respective labeling. Then, we can define an anomaly pattern as the respective label set and its corresponding (induced) subgraph, covering a subset of nodes and set transitions, respectively. In that way, we can not only include anomalous episodes in the sequential anomaly detection step, but we can include descriptive information for enabling further inspection, explanation and/or exemplification [10, 15] by the operator or the process engineer. Thus, the descriptive

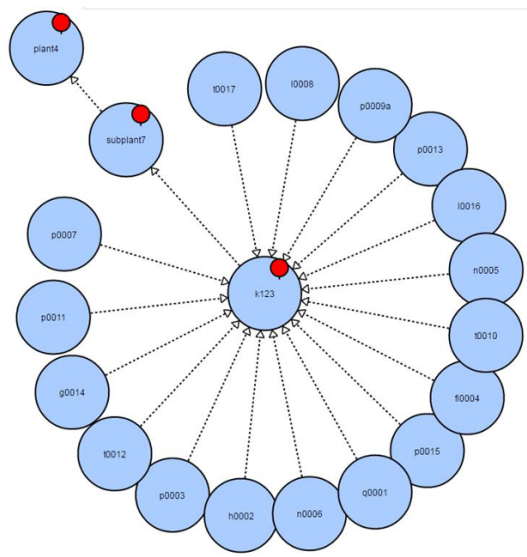


Figure 3: Example of a (conceptual) knowledge graph [14].

[1] https://github.com/rapidminer/rapidminer-extension-hypgraphs/releases

Figure 4: Larger (anonymized) example of a knowledge graph.

features can, e.g., indicate important indicators for anomalies like sensors or alarm related labels as proxies for specific faults. The descriptive information can, for example, be derived by the integration of unstructured information such as plant topology information derived from a Knowledge Graph [14]. Such a graph can be constructed from heterogeneous information, such as sensor measurements, alarm logs, process and instrumentation diagrams (P&IDs), shift books, operation manuals etc. Figure 3 shows an abstracted example of a knowledge graph showing the conceptual plant-subplant relations and measurements [14], while Figure 4 shows an anonymized example of a larger plant context.

2) *Anomaly Detection on Metric Data*

For detecting outliers on the numeric sensor data we apply the Local Outlier Factor (LOF) algorithm by Breunig et al. [18], as implemented in the Anomaly Detection extension [4, 31] for RapidMiner. The algorithm estimates local deviations of the data points using a defined distance function. It compares the local density of a point to the density of its *k* nearest neighbors. Due to the nature of the provided sensor data, the concept of a locally sensitive algorithm is useful, because with different set points (for plant operation) range and characteristics of the sensor readings can vary greatly. The outlier scores can be calculated for either all available sensors, for certain subgroups, or single sensors, depending on the desired granularity.

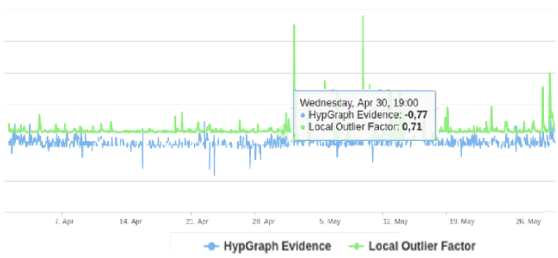


Figure 5: Overview of outlier scores and HypGraphs evidence values. A high outlier score indicates anomalous sensor readings, while a low the evidence score indicates deviating alarm sequences

IV. Process Model & Implementation

Distributed storage and computation systems, can handle the evaluation of several years of production data. In the context of the FEE project we want to build upon a two-layered computation architecture to enable the plant operators and system engineers to design and test their processes on a local machine and execute memory and computational intensive processes in the Hadoop infrastructure [19, 24].

The first part of the analytical workflow is to build the transition network for training and testing the hypotheses. We build these hypotheses on real plant data and calculate the transition matrices for hourly time slots over a period of two months. In the same way, after further preprocessing (smoothing and down-sampling) we aggregate the raw sensor data. The calculated outlier score is shown in Figure 5, together with the evidence scores. A high outlier score indicates possible anomalous sensor readings and a low evidence score indicates deviating transition patterns in the alarm sequences. As one can see, the two values are apparently not strongly correlated. But this shows, that the two algorithms monitor different aspects of the plant behavior: the lowest evidence scores occur in the early morning hours with nearly no alarm transitions (when normally several dozen alarms are recorded per hour).

For further inspecting the outlier scores, we have built another dashboard. This shows the *k* highest outlier score for single sensor readings for a selected time segment, for example by clicking on a specific time point in the dashboard from Figure 5. In addition, this board shows the associated sensor readings. With this drill-down from a high level of abstraction for a whole processing unit down to single sensor readings, a process engineer is able to identify and inspect possible critical situations in a convenient way.

V. Conclusion and Future Work

This paper presented a sequential modelling and anomaly analytics approach in an industrial application context. Based on first order Markov chain models and methods for modelling and comparing networks and transition matrices, we sketched an approach for comparing hypotheses with observed "reference" sequential patterns. Furthermore, we described the extension to integrating descriptive information in the sequential modeling approach. In addition, we also showed a comparison between our approach and an established outlier detection algorithm. It became evident that both methods target different aspects of detecting anomalous behavior.

For future work, we aim at extending the proposed approach by integrating the knowledge gained from the conceptual knowledge graph, e.g., by grouping and analyzing the outlier scores for the sensors associated with specific functional units. We also plan to integrate the system into the Big data architecture proposed in [24]. As outlined above, we want to extend that two-layered computation architecture for enabling flexible and powerful Big data processing approaches, also including large-scale descriptive subgroup [26], sequence [38, 43], and graph mining methods [13] for efficient exceptional model mining and anomaly analytics.

References

[1] Aalst, W.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, Berlin (2011)

[2] Abele, L., Anic, M., Gutmann, T., Folmer, J., Kleinsteuber, M., Vogel-Heuser, B.: Combining Knowledge Modeling and Machine Learning for Alarm Root Cause Analysis. In: Proc. IFAC Volumes, 46(9):1843–1848. International Federation of Automatic Control (2013)

[3] Akoglu, L., Tong, H., Koutra, D.: Graph Based Anomaly Detection and Description. Data Min Knowl Disc 29(3), 626–688 (May 2015)

[4] Amer, M., Goldstein, M.: Nearest-Neighbor and Clustering-based Anomaly Detection Algorithms for Rapidminer. In: Proc. of the 3rd RapidMiner Community Meeting and Conference (RCOMM 2012). pp. 1–12 (2012)

[5] Atzmueller, M.: Analyzing and Grounding Social Interaction in Online and Offline Networks. In: Proc. ECML PKDD. LNCS, vol. 8726, pp. 485–488. Springer, Heidelberg, Germany (2014)

[6] Atzmueller, M.: Data Mining on Social Interaction Networks. Journal of Data Mining and Digital Humanities 1 (June 2014)

[7] Atzmueller, M.: Subgroup Discovery - Advanced Review. WIREs: Data Mining and Knowledge Discovery, (5)1:35—49 (2015)

[8] Atzmueller, M.: Detecting Community Patterns Capturing Exceptional Link Trails. Proc. IEEE/ACM ASONAM, IEEE Press, Boston, MA, USA (2016)

[9] Atzmueller, M.: Local Exceptionality Detection on Social Interaction Networks. In: Proc. ECML PKDD 2016: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. Springer, Heidelberg, Germany (2016)

[10] Atzmueller, M., Baumeister, J., Puppe, F.: Introspective Subgroup Analysis for Interactive Knowledge Refinement. In: Proc. FLAIRS Conference, pp. 402-407, AAAI Press, Palo Alto, CA, USA (2006)

[11] Atzmueller, M., Doerfel, S., Mitzlaff, F.: Description-Oriented Community Detection using Exhaustive Subgroup Discovery. Information Sciences, 329, 965-984. (2016)

[12] Atzmueller, M., Doerfel, S., Hotho, A., Mitzlaff, F., Stumme, G.: Face-to-Face Contacts at a Conference: Dynamics of Communities and Roles. In: Modeling and Mining Ubiquitous Social Media, LNAI, vol. 7472. Springer, Heidelberg, Germany (2012)

[13] Atzmueller, M., Mollenhauer, D., Schmidt, A.: Big Data Analytics Using Local Exceptionality Detection. In: Enterprise Big Data Engineering, Analytics, and Management, IGI Global, Hershey, PA, USA, 2016.

[14] Atzmueller, M., Kloepper, B., Mawla, H.A., Jäschke, B., Hollender, M., Graube, M., Arnu, D., Schmidt, A., Heinze, S., Schorer, L., Kroll, A., Stumme, G., Urbas, L.: Big Data Analytics for Proactive Industrial Decision Support: Approaches First Experiences in the Context of the FEE Project. atp edition 58(9):62-74 (2016)

[15] Atzmueller, M., Roth-Berghofer, T.: The Mining and Analysis Continuum of Explaining Uncovered. Proc. 30th SGAI International Conference on Artificial Intelligence (2010)

[16] Atzmueller M, Schmidt A, Kibanov M. DASHTrails: An Approach for Modeling and Analysis of Distribution-Adapted Sequential Hypotheses and Trails. In: Proc. WWW 2016 (Companion). IW3C2 / ACM, New York, NY, USA (2016)

[17] Atzmueller M., Schmidt A., Kloepper B., Arnu D.: HypGraphs: An Approach for Modeling and Comparing Graph-Based and Sequential Hypotheses. In: Proc. ECML PKDD Workshop on New Frontiers in Mining Complex Patterns (NFMCP). Riva del Garda, Italy (2016).

[18] Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: OPTICS-OF: Identifying Local Outliers, pp. 262–270. Springer, Berlin/Heidelberg (1999)

[19] Dean, J., Ghemawat, S.: Mapreduce: Simplified Data Processing on Large Clusters. Communications of the ACM 51(1), 107–113 (2008).

[20] Folmer, J., Kirchen, I., Trunzer, E., Vogel-Heuser, B., P□tter, T., Graube, M., Heinze, S., Urbas, L., Atzmueller, M., Arnu, D: Challenges for Big and Smart Data in Process Industries. *atp edition*, 01/02 (2017)

[21] Folmer, J., Schuricht, F., Vogel-Heuser, B.: Detection of Remporal Dependencies in Alarm Time Series of Industrial Plants. Proc. IFAC, pp. 24–29, International Federation of Automatic Control (2014)

[22] Hawkins,D.: Identification of Outliers. Chapman and Hall, London, UK (1980)

[23] Kibanov, M., Atzmueller, M., Scholz, C., Stumme, G.: Temporal Evolution of Contacts and Communities in Networks of Face-to-Face Human Interactions. Science China Information Sciences 57 (2014)

[24] Klöpper, B., Dix, M., Schorer, L, Ampofo, A., Atzmueller, M., Arnu, D., Klinkenberg, R.: Defining Software Architectures for Big Data Enabled Operator Support Systems. In: Proc. INDIN. IEEE Press, Boston, MA, USA (2016)

[25] Leman, D., Feelders, A., Knobbe, A.: Exceptional Model Mining. In: Proc. ECML PKDD, pp. 1-16, Springer, Heidelberg, Germany (2008)

[26] Lemmerich, M., Atzmueller, M., Puppe, F.: Fast Exhaustive Subgroup Discovery with Numerical Target Concepts. Data Mining and Knowledge Discovery, (30):711-762 (2016)

[27] Lemmerich, M., Becker, M., Atzmueller, M.: Generic Pattern Trees for Exhaustive Exceptional Model Mining. Proc. ECML PKDD 2012, pp. 277-292, Springer, Heidelberg, Germany (2012)

[28] Lempel, R., Moran, S.: The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect. Computer Networks 33(1), 387–401 (2000)

[29] Macek, B.E., Scholz, C., Atzmueller, M., Stumme, G.: Anatomy of a Conference. In: Proc. ACM Hypertext. pp. 245–254. ACM Press, New York, NY, USA (2012)

[30] Martí, L., Sanchez-Pi, N., Molina, J.M., Garcia, A.C.B.: Anomaly Detection based on Sensor Data in Petroleum Industry Applications. Sensors 15(2), 2774–2797 (2015)

[31] Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: Yale: Rapid Prototyping for Complex Data Mining Tasks. In: Proc. KDD. pp. 935–940. ACM, New York, NY, USA (2006)

[32] Mitzlaff, F., Atzmueller, M., Benz, D., Hotho, A., Stumme, G.: Community Assessment using Evidence Networks. In: Analysis of Social Media and Ubiquitous Data. LNAI, vol. 6904 (2011)

[33] Mitzlaff, F., Atzmueller, M., Hotho, A., Stumme, G.: The Social Distributional Hypothesis. SNAM 4(216) (2014)

[34] Munoz-Gama, J., Carmona, J., van der Aalst, W.M.P.: Single-Entry Single-Exit Decomposed Conformance Checking. Inf. Syst. 46, 102–122 (2014)

[35] Pirolli, P.L., Pitkow, J.E.: Distributions of Surfers' Paths Through the World Wide Web: Empirical Characterizations. WWW 2(1-2) (1999)

[36] Ranshous, S., Shen, S., Koutra, D., Harenberg, S., Faloutsos, C., Samatova, N.F.: Anomaly Detection in Dynamic Networks: A Survey. WIREs: Comput. Statistics 7(3), 223–247 (2015)

[37] Rozinat, A., Aalst, W.: Conformance Checking of Processes Based on Monitoring Real Behavior. Information Systems 33(1), 64–95 (2008)

[38] Seipel, D., K□hler, S., Neubeck, P., Atzmueller, M.: Mining Complex Event Patterns in Computer Networks. In: New Frontiers in Mining Complex Patterns (NFMCP), Springer, Heidelberg, Germany (2013)

[39] Singer, P., Helic, D., Hotho, A., Strohmaier, M.: Hyptrails: A Bayesian Approach for Comparing Hypotheses about Human Trails. In: Proc. WWW. ACM, New York, NY, USA (2015)

[40] Singer, P., Helic, D., Taraghi, B., Strohmaier, M.:Memory and Structure in Human Navigation Patterns. PLoS ONE 9(7) (2014)

[41] Strelioff, C.C., Crutchfield, J.P., Hübler, A.W.: Inferring Markov Chains: Bayesian Estimation, Model Comparison, Entropy Rate, and Out-of-Class Modeling. Physical Review E 76(1), 011106 (2007)

[42] Vogel-Heuser, B., Schütz, D., Folmer, J.: Criteria-based alarm flood pattern recognition using historical data from automated production systems (aps). Mechatronics 31, 89–100

[43] Weiss, C. H., Atzmueller, M.: EWMA Control Charts for Monitoring Binary Processes with Applications to Medical Diagnosis Data. Qual. Reliab. Engng. Int., 26: 795–805 (2010)

Semantically Annotated Manufacturing Data to support Decision Making in Industry 4.0: A Use-Case Driven Approach

Bringing new Structure into existing Data to support Smart Manufacturing

Stefan Schabus
Infineon Technologies Austria AG,
Villach - Austria

Johannes Scholz
Graz University of Technology,
Institute of Geodesy, Research Group Geoinformation
Graz - Austria

***Abstract*—Smart Manufacturing or Industry 4.0 is a key approach to increase productivity and quality in industrial manufacturing companies by automation and data driven methods. Smart manufacturing utilizes theories from cyber-physical systems, Internet of Things as well as cloud computing. In this paper, the authors focus on ontology and (spatial) semantics that serve as technology to ensure semantic interoperability of manufacturing data. Additionally, the paper proposes to structure production relevant data by the introduction of geography and semantics as ordering dimensions. The approach followed in this paper stores manufacturing data from different IT-systems in a graph database. During the data integration process, the system semantically annotates the data – based on an ontology, developed for that purpose - and attaches spatial information. The approach presented in this paper facilitates an analysis of manufacturing data in terms of semantics and the spatial dimension. The methodology is applied to two use-cases of a semiconductor manufacturing company. The first use-case deals with the data analysis for incident analysis utilizing semantic similarities. The second use-case supports decision making in the manufacturing environment, by the identification of potential bottlenecks in the semiconductor production line.**

Keywords—Semantic Data; Smart Manufacturing; Industry 4.0; Spatial Data; Geography

I. INTRODUCTION

Manufacturing industry needs to focus on the competitiveness of manufacturing processes, as global markets tend to be increasingly competitive. Therefore, any manufacturing company works on strategies to increase productivity, efficiency, performance and to realize cost-savings [1; 2]. To achieve these long-term goals, IT-technology has been identified as promising "tool". Utilizing digital technology in the manufacturing sector leads to Industry 4.0 or smart manufacturing activities. Digitalization of manufacturing processes makes use of cyber-physical systems, Internet of Things and cloud computing [3; 4; 5].

Currently, huge amounts of data are generated, e.g. by sensors and manufacturing devices, that are intended to keep track of processes, quality issues and also transportation processes of production assets. Due to the fact, that a great variety of proprietary IT-systems are present in manufacturing enterprises, the collected datasets can hardly be combined in an interactive manner. The variety of IT-systems is currently justified, because each system serves a certain purpose – and is often required by a single department. In recent times, the necessity to make an integrated analyses of collected datasets is growing, partly based on Industry 4.0 initiatives. Hence, the lack of existing interoperability to share and integrate datasets within manufacturing companies becomes evident. Interoperability in this context is regarded as semantic interoperability. While syntactic interoperability can be easily achieved with (spatial) ETL tools, semantic interoperability enables an ad-hoc sharing and analysis of different datasets.

To utilize the potential of the collected manufacturing data, this paper proposes to add semantic annotations and geographical information, and use these added information layers for analysis and decision making purposes. Therefore, the indoor manufacturing space and the manufacturing processes need to be semantically described. Ontologies are an appropriate way to formally describe a universe of discourse [6; 7; 8]. Ontologies consist of entities, relations connecting the entities and rules [9]. Recently, the modeling of cyber-physical systems proved successful [12]. Papers [10; 11] elaborate on the spatial dimension of ontologies. In the context of geography and indoor space, ontologies are used to model indoor space [13; 14; 15]. To utilize ontologies in an IT architecture, the concept of Semantic Web offers the possibility to share data and their meaning – i.e. their semantics [16]. Semantic methodologies have been recently applied in smart manufacturing environments [4; 17; 18]. These approaches mostly utilize semantic web services to share semantics and/or semantic annotations of manufacturing data.

The research question of this paper focuses on the integration of heterogeneous manufacturing datasets with the help of ontologies and the geographical dimension. Additionally, the paper elaborates on the question if semantics and geography can

P. Haber et al. (Hrsg.), *Data Science – Analytics and Applications*, http://doi.org/10.1007/978-3-658-19287-7_14

help to analyze manufacturing datasets and in turn support decision-making.

The methodology in this paper is as follows. First, we develop an ontology for manufacturing purposes that is based on an indoor navigation ontology [13]. This ontology describes manufacturing processes, production assets, manufacturing devices, as well as indoor manufacturing environments (i.e. cleanroom). In addition, the ontology includes geographical phenomena of the entities, such as positions of production assets over time, as well as a topological model of the indoor space – for routing and navigation purposes. Based on the developed ontology, heterogeneous datasets are integrated in a Graph database. The integration process includes a semantic annotation of the datasets and the establishment of typed links between abstract classes and entities, as well as between entities. To justify the approach, we evaluate the potential of semantically annotated and geographically amended data in two use-cases. The use-cases are located in a semiconductor manufacturing company, a highly flexible and complex manufacturing environment. Use-case #1 deals with incident analysis in the manufacturing line and the search for similar products that are potentially damaged. Use-case #2 deals with identifying bottlenecks – in terms of manufacturing capacity – in the semiconductor manufacturing line under review.

As stated before, both use-cases are located in a semiconductor manufacturing facility with cleanroom restrictions. The following paragraph is intended to give a brief overview of the semiconductor manufacturing facility. Each production asset requires several hundred manufacturing steps from raw silicon wafer to the final microchip. Each step can be processed by several devices, which may be geographically dispersed over the production facility. In the facility, assets with different degrees of completion are present at the same time. In addition, several hundred different products or product types are manufactured simultaneously in the same facility. Each single production asset may have varying manufacturing time – lasting from several days up to a couple of weeks. Furthermore, the proportion of products and the overall manufacturing quantity is changing on a weekly basis, depending on the customer needs. This flexible production is the reason for the absence of a conveyor belt. Thus, production assets are mainly transported on trolleys from one production step to the next one. Further details of the universe of discourse are to be found in literature [13; 19; 20].

The reminder of the paper is organized as follows: Section 2 elaborates on semantic annotated manufacturing data and the modeling aspects thereof. In addition, the storage in a graph database is described. Section 3 focuses on the analysis of manufacturing data and the data's semantics based on two selected use-cases: #1) incident analysis in the manufacturing line and #2) potential bottleneck identification. Section 4 is a conclusion and discusses potential future research directions of smart manufacturing based on semantically annotated manufacturing data.

II. Modelling, Storing and Publishing Semantically Annotated Manufacturing Data

The following section elaborates on the modelling aspects of the universe of discourse. Furthermore, we focus on the data storage in a spatial graph database with additional semantics. Finally, the visualization of semantic annotated data in the spatial graph database and the querying thereof are highlighted. The data model in a graph database follows a graph-oriented structure. Due the fact that the data are machine readable – and may be shared as Resource Description Framework (RDF) [21] – the approach ensures semantic and syntactic interoperability.

A. Spatial-temporal Ontology to model Manufacturing Environment

An ontology in the context of knowledge sharing "is a specification of a conceptualization" [8]. Thus, an ontology is a description of the concepts and relationships that can exist in a universe of discourse [6]. [10] and [22] describe the modeling of the spatial domain with the help of ontologies. As ontologies describe the universe of discourse in a formal way, they can help to foster semantic interoperability [23; 24].

The ontology developed in this context is based on an Indoor Navigation Ontology in a production environment [13]. After a review of the existing Navigation Ontology, some additional entities and relationships are added - for the purpose of modeling manufacturing processes accurately. Figure 1 shows the ontology published by [13] with the most important amendments made. Oval shapes represent classes, and relationships are illustrated with solid arrows. The root element – thing – has several top-level entities like Navigation Agents (i.e. production assets), Corridors, Graph, Production Unit.

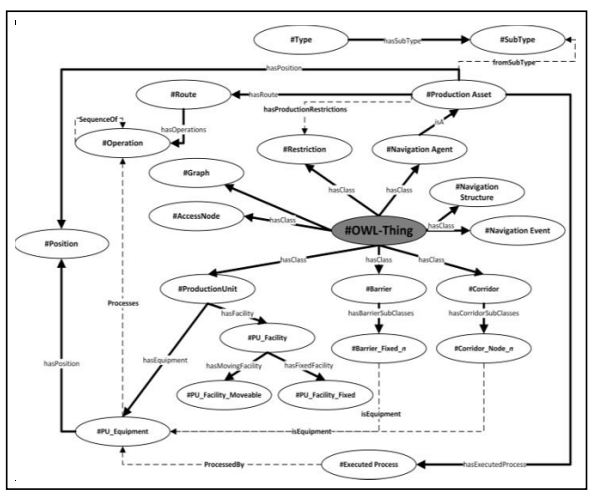


Fig. 1. Snapshot of the adapted ontology by [8] showing class hierarchies as solid arches and added semantics as dashed arches.

The class corridors represents the corridors in the production hall, which have several corridor nodes – each one having a geographical position. Each corridor node is connected with a piece of manufacturing equipment. The class graph denotes the

underlying geographical structure necessary for routing and navigation purposes. The production assets are categorized in types and sub-types, besides a geographical position. The class route – connected with the production asset – holds information on the sequence of manufacturing operations (planned and historic operations). In addition, each manufacturing equipment is able to process 1-n operations. A production unit is either a facility – like a table or shelf – or a manufacturing equipment. Facilities are moveable or fixed. In addition, figure 1 shows that movement barriers exist in the manufacturing environment having several sub-classes, such as fixed barriers. The dashed arrows represent added relationships, like that the class "Barrier_fixed_*n*" may be an equipment.

The geographical information, present in the model at hand, is stored in the classes position and graph. The class position contains points with an ordered pair of x/y coordinates amended with the floor information. Each piece of equipment, production facility and each production asset have a geographical position. In addition, the model offers the possibility to store historical positions of each entity – resulting in a trajectory for each aforementioned entity. The entity graph represents a routable network consisting of vertices and edges, connecting entities in the indoor space (i.e. manufacturing devices, shelves, access nodes, etc.). Each vertex of the network has a specific position, which makes use of the position entity – i.e. having an x/y coordinate.

The data model contains a temporal component as well. Each production asset has an associated planned manufacturing route – in this context a sequence of manufacturing operations. In addition, the ontology allows the storage of the historic manufacturing operations in a temporal order, with the help of a temporal ordering.

B. Storage and Analysis in a Graph Database

A graph database is a database management system that is capable of creating, reading, updating, and deleting data in form of graphs in a database. A general introduction of graph database and their fundamental concepts are presented in [25]. The combination of ontologies and graph databases in the field of Geographic Information Science are published in [26]. Graphs are a collection of ordered pairs of vertices and edges. In this context, vertices correspond with entities and relationships connect entities – thus corresponding with edges. This allows the modelling of real world in terms of graphs, e.g. supply-chains, road network, or medical history for populations (see e.g. [27]). They became very popular due to their suitability for social networking, and are used in systems like Facebook Open Graph, Google Knowledge Graph, or Twitter FlockDB [27].

To make use of the developed ontology for data analysis purposes, the authors migrate the ontology – stored as Ontology Web Language (OWL) - into the graph database. Hence, the abstract entity classes and their relationships are part of the graph database. Subsequently, relevant manufacturing data are migrated into the graph database and semantically annotated – to open up the possibility for semantic data analysis. The following datasets are migrated into the graph database (see Figure 2):

- Manufacturing data
    - Historic manufacturing data of production assets including quality related data
    - Planned manufacturing operations for each production asset
- Spatial data: i.e. routable graph structure, manufacturing equipment positions, trajectories of production assets.
- Attributive information: e.g. attributes of production assets, equipment, spatial/temporal movement restrictions;

The migration of the data sets into the graph database is a process that uses spatial and non-spatial data from object-relational database management systems (DBMS) and other proprietary IT-systems (e.g. Computer Aided Design Systems). The process analyzes the data with respect to the developed ontology. Thus, we identify individuals of abstract entities, create the relationships as typed (annotated) edges based on the ontology, and assign the properties of each individual. An example is the analysis of historical manufacturing data. Each individual production asset and its trajectory is queried in the corresponding object relational DBMSs. For each production asset, the process creates a new vertex in the graph database, with a typed (semantically annotated) relationship to the abstract entity 'Production Asset'. Future processing information is added via linked operations, which are executable at a certain equipment. Historic manufacturing data, describing the asset history, like executed processes, processing equipment and the asset's trajectory are stored as well. The trajectory is a sequence of asset positions – where the positions are ordered by time.

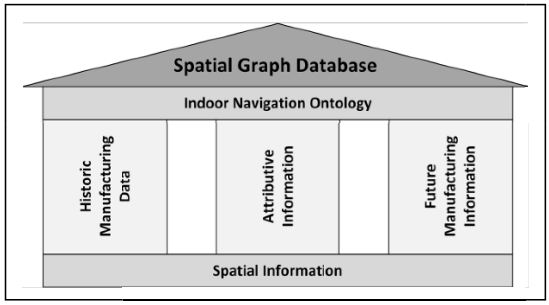


Fig. 2. Pillars of the spatial graph database that serves as data pool for semantically annotated historic, attributive, future and spatial data.

Figure 3 shows an example visualization of the spatial graph database, which is implemented utilizing the graph database 'Neo4j'. The visualization capabilities of Neo4j allow the displaying of graph elements. A single piece of equipment – i.e. an individual – can be visualized including the relationships. In Figure 3 an equipment is depicted that is linked to attributes such as name, location, or possible processes. In addition, this specific equipment is a barrier for the transportation of production assets.

Figures 4 illustrates two production assets and their relationships in the graph database. In this example, it is clearly visible that each asset is processed at specific pieces of equipment. In order to analyze this fact with the help of algorithms, any graph database supports the application of algorithms originating from graph theory.

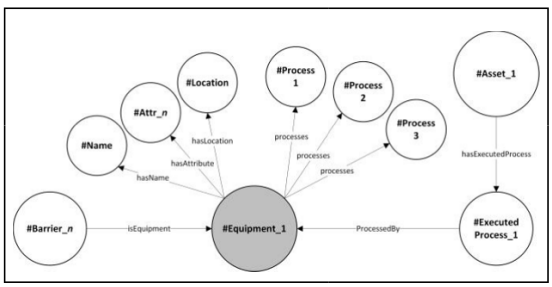


Fig. 3. Visualization of the spatial graph database with the equipment as a center point. It includes historic information via the executed process, attributive information via hasName or hasAttribute and future information via possible executable processes at the equipment.

A detailed analysis of the data present in the graph database is based on graph theoretical algorithms, such as breadth-first search and shortest-path algorithms, in conjunction with the inherent semantics and geographical information. Due to the fact, that the edges are semantically annotated, any graph analysis algorithm can make use of that information. An example is the dynamic identification of existing paths between entities (i.e. vertices) that supports the analysis of similarities of entities. Such similarities may be: similar location, similar processing equipment, or similar trajectories. In figure 4, an example for the similarity evaluation of two production assets is given. Asset 1 and 2 have a variety of attributes, and a number of processing equipment but only one path from asset 1 to 2 exists via equipment 3 – exploiting the "*processedby*" relationship. Hence, equipment 3 processed both assets. By adding the temporal dimension, incident analysis becomes possible. Incident analysis tries to identify potentially affected production assets of an incident – like an equipment failure, contamination issue or cleanroom problem. Therefore, the analysis of an assets' position at each time instant is inevitable, to identify which assets have been present in the affected area at a specific time (interval).

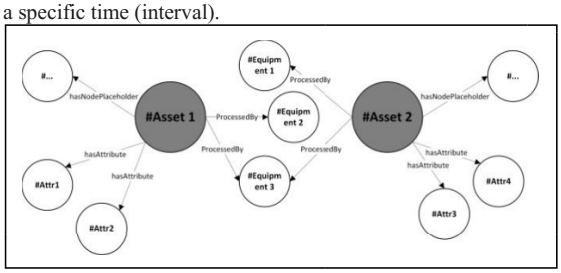


Fig. 4. Example showing the data present in the graph database. If a path along the 'ProcessedBy'-edges from asset 1 to asset 2 via equipment 3 is possible, we can conclude that both were processed by equipment 3

III. Analysis of Manufacturing Data based on Semantically annotated Data: Use-Cases

The analysis of semantically annotated manufacturing data is shown based on two use-cases. Both are based on the developed ontology and manufacturing data present in the graph database. The first use-case deals with incident analysis, whereas the second use-case elaborates on the identification of possible manufacturing bottlenecks. Both use-cases rely on historic manufacturing data, which are structured and analyzed to identify similar movement patterns and similarities regarding the utilized equipment.

A. Incident analysis – Identification of similarities with respect to Space and Time

An incident is an "an unplanned, undesired event that hinders completion of a task and may cause injury or other damage" [28]. Incidents may include minor human operator injury, minor damage to smaller system parts, or failure of a component – but do not disrupt the system as a whole [29]. In the context of this paper, incident analysis is regarded as the process of finding production assets that are similar to assets having quality issues. An incident in this respect can be an equipment failure, contamination or cleanroom problem. In general, only a sample of the production assets have to undergo a full quality check at the end of the production line. Hence, if quality anomalies exist, similar production assets need to be identified and quality checked in order to minimize the possibility of delivering defective products.

The incident analysis highlighted in this use-case, elaborates on an analysis with the help of spatial queries and the determination of similarities on the level of production assets. Based on a defective production asset, an incident – here a malfunctioning air cleaning system over a certain time span – is identified as root cause. This incident might lead to contamination issues on production assets. Figure 5 depicts the occurred incident in the manufacturing environment with a red circle. The affected area is visualized as a circular red circular object in Figure 5. In order to limit the complexity of the semantic query, the production assets traversing the incident area are selected. Therefore, the trajectories of the assets are analyzed, which is depicted in Figure 5.

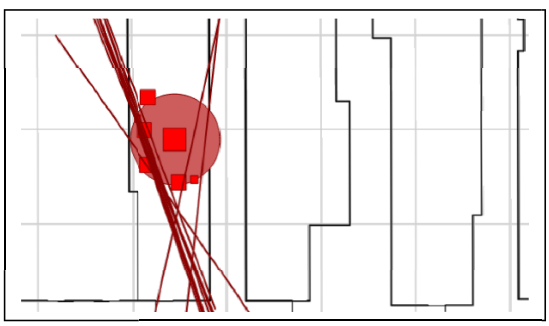


Fig. 5. Trajectories of production assets crossing an incident area. The area is limited by space and time. The intersecting lines indicate assets that have been potentially affected by the incident, which is marked with a big red circle.

The spatial query in the graph database is as follows: '*spatial.intersects('layer_tracked_positions_', Incident_Area) YIELD node_asset RETURN node_asset, node_geometry'* and returns the asset identifier and asset geometry to be mapped. In addition, this result set is further reduced by duration of the incident. The resulting set is the starting point for the semantic analysis. The spatial-temporal selection reduces the number of potentially affected assets from of a few thousand to a few hundred.

In this use-case, the defective asset serves as reference asset for identifying similar possibly affected assets. Thus, Figure 6 shows the methodology of identifying assets using the spatial graph database and semantics. We have the possibility to identify similarities between the faulty asset and potentially affected assets, by analyzing the graph. In Figure 6, the possible paths from the affected asset 1 to potentially affected assets 2 and 3 are visualized. The analysis makes use of the graph structure, and calculates the path length between the assets strictly allowing edges annotated as "*fromType*" and "*hasSubType*". Hence, asset 1 and 2 are similar because the shortest path between them comprises four edges – both have a same product type. Asset 3 has a different product type and subtype resulting lower similarity – so there is no connectivity between asset 1 and 3. In Figure 6 the pseudocode of the analysis in the graph database is depicted.

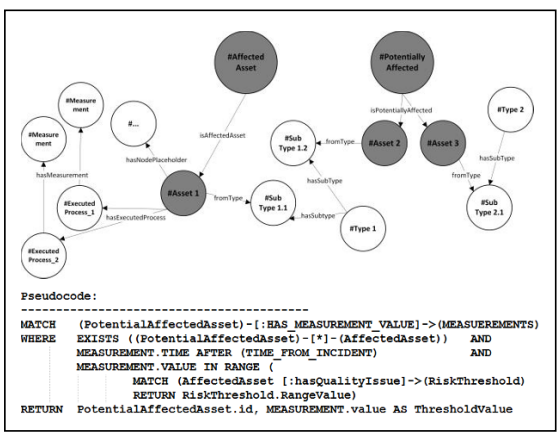


Fig. 6. Identification of similarites between an identified affected asset and potentially affected assets using the graph database and semantics. A connecting path between asset 1 and 2 is possible due to similar product types. The pseudo code shows an evaluation of existing measurement values of the potentially affected asset.

B. Identification of Potential Bottlenecks based on Historic Data

The identification of potential bottlenecks is a crucial task for the manufacturing line management, as bottlenecks may negatively influence the efficiency of the production line. In this paper, we identify bottlenecks based on the geographical information associated with the production assets. We analyze the trajectories of production assets and potential bottlenecks.

Bottlenecks in a production line are present if the capacity of manufacturing equipment for a specific task is lower than the inflow of assets to be processed. Thus, assets need to be stored in shelves, and have to "wait" until they can be processed. The trajectories of assets are an indicator for detecting bottlenecks. An evaluation of the movement of assets – i.e. distance from one position to the next position – gives an indication on the potential bottlenecks present in a manufacturing line. If an asset does not move over a certain period of time it regarded as being waiting.

To determine assets with a "high" waiting time, we use the following methodology. The waiting time of an asset is regarded as timespan between being placed in a shelf, and the start of the next manufacturing operation. First, we calculate an average waiting time and standard deviation over the last month for each asset type, per operation and equipment. This number serves as indication for the "normal" waiting time in the factory. Thus, we calculate the recent waiting time for each asset type, per operation and equipment. This is done using a moving average waiting time calculation, with a two-hour window. Assets having a higher waiting time than the 2-sigma range of the normal waiting time are classified as "delayed". By exploiting the geographical information of each asset, it is possible to identify clusters of delayed production assets – i.e bottlenecks – for different asset types, operations, or pieces of equipment. In addition, an analysis of historical bottlenecks may reveal spatial-temporal patterns of a "congested" production line that could be of interest for factory managers.

Figure 7 shows the pseudo code for the proposed analysis. The code identifies all waiting times for similar assets by selecting one asset of interest and identifying therefore all similar assets. The result can be visualized as a heatmap depicting bottlenecks in the indoor manufacturing environment.

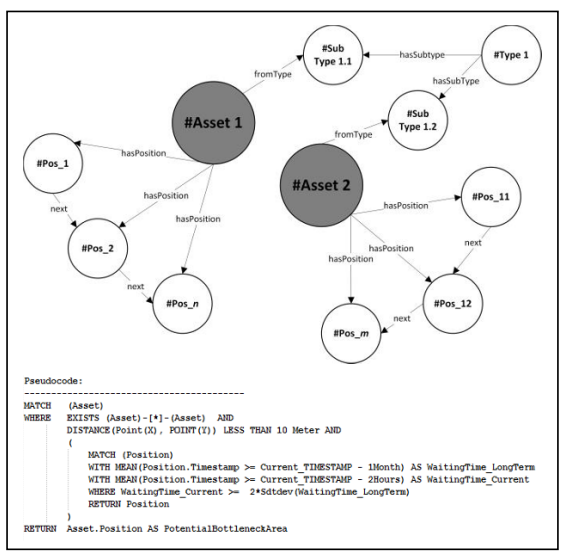


Fig. 7. Identification of bottlenecks of production assets based on the waiting time.

IV. Conclusion and Discussion

To conclude, the paper elaborates on the creation and analysis of semantically annotated manufacturing data. Based on two use-cases we show that semantics and geographical information can contribute to improving decision making in manufacturing environments. Both use-cases rely on data analysis of semantically annotated (spatial) data in a graph database. Semantic annotations in the graph database are a result of an ontology, describing the indoor semiconductor-manufacturing environment.

The data analysis in both use-cases reveals the potential of a combined geographical and semantical data analysis. In addition, the integration of various data sources in a graph database and their semantic annotation serve as basis for the use-cases discussed in the paper. This underpins the argument to strengthen semantic interoperability – also in manufacturing companies. Hence, intelligent data sharing and publishing strategies like Linked Data [16] could be an appropriate strategy for manufacturing companies. A Linked Data approach within a factory could open up the possibility to semantically query available datasets using SPARQL. Additionally, the geographical domain could be considered by using the query language GeoSPARQL [30].

V. References

[1] J. Davis, T. Edgar, J. Porter, J. Bernaden and M. Sarli, "Smart manufacturing, manufacturing intelligence and demand-dynamic performance," Computers & Chemical Engineering, vol. 47, 2012, pp. 145-156.

[2] R. H. Nystr□m, I. Harjunkoski and A. Kroll, "Production optimization for continuously operated processes with optimal operation and scheduling of multiple units," Computers & chemical engineering, vol. 30, no. 3, 2006, pp. 392-406.

[3] M. Hermann, T. Pentek, T. and B. Otto, "Design Principles for Industry 4.0 Scenarios" in 49th Hawaii International Conference on System Sciences (HICSS), 2016, pp. 3928-3937.

[4] D. Zuehlke, "SmartFactory—Towards a factory-of-things," Annual Reviews in Control, vol. 34, 2010, pp. 129-138.

[5] J. Lee, H.-A. Kao, and S. Yang, "Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment," Procedia CIRP, vol. 16, 2014, pp. 3-8.

[6] M. Uschold and M. Gruninger, "Ontologies: Principles, methods and applications," Knowledge engineering review, vol. 11, no. 2, 1996, pp. 93-136.

[7] B. Smith, "Objects and their environments: from Aristotle to ecological ontology," The life and motion of socio-economic units: GISDATA, vol. 8, 2001, pp. 79-97.

[8] T. R. Gruber, "A translation approach to portable ontology specification," Knowledge Acquisition, vol. 5, no. 2, 1993, pp. 199-220.

[9] E. Davis, Reprsentations of commonsense knowledge, Morgan Kaufmann, 1990.

[10] Y. Bishr and W. Kuhn, "Ontology-based modelling of geospatial information," in 3rd AGILE Conference on Geographic Information Science, May, 2000, pp. 25-27.

[11] P. Grenon and B. Smith, "SNAP and SPAN: Towards dynamic spatial ontology," Spatial cognition and computation, vol. 4, no. 1, 2004, pp. 69-104.

[12] P. Derler, E. A. Lee, and A. S. Vincentelli, "Modeling cyber-physical systems," Proceedings of the IEEE, vol. 100, 2012, pp. 13-28.

[13] J. Scholz and S. Schabus, "An indoor navigation ontology for production assets in a production environment", Geographic Information Science: 8th International Conference, GIScience 2014, Vienna, Austria, September 24-26, 2014. Proceedings," M. Duckham, E. Pebesma, K. Stewart and A. U. Frank, Eds., Ch, a: Springer International Publishing, 2014, pp. 204-220.

[14] L. Yang, and M. A. Worboys, "A navigation ontology for outdoor-indoor space:(work-in-progress)," Proceedings of the 3rd ACM SIGSPATIAL international workshop on indoor spatial awareness, 2011, pp. 31-34,

[15] M. Raubal, and M.A.Worboys, "A formal model of the process of wayfinding in built environments Spatial information theory," Cognitive and computational foundations of geographic information science, Springer, 1999, pp. 381-399.

[16] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data-the story so far," Semantic Services, Interoperability and Web Applications: Emerging Concepts, 2009, pp. 205-227.

[17] K. Jung, K. Morris, K. W. Lyons, S. Leong, and H. Cho, "Mapping strategic goals and operational performance metrics for smart manufacturing systems," Procedia Computer Science, vol. 44, 2015, pp. 184-193.

[18] L. M. S. De Souza, P. Spiess, D. Guinard, M. K□hler, S. Karnouskos, and D. Savio, "Socrades: A web service based shop floor integration infrastructure" The internet of things, Springer, 2008, pp. 50-67.

[19] S. Schabus, and J. Scholz, "Geographic Information Science and technology as key approach to unveil the potential of Industry 4.0: How location and time can support smart manufacturing," in 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO), 2015.

[20] S. Schabus, J. Scholz, and A. Skupin, "Spatial-temporal Patterns of Production Assets in an Indoor Production Environment," in *Proceedings of Workshop" Analysis of Movement Data©14" Workshop at GIScience 2014*, 2014.

[21] Suvee D. "Visualizing RDF Schema inferencing through Neo4J, Tinkerpop, Sail and Gephi", Web: http://datablend.be/?p=260, 2011, [last visited 20-12-2016].

[22] A.U. Frank, "Tiers of ontology and consistency constraints in geographical information systems," International Journal of Geographical Information Science, vol. 15, no. 7, 2001, pp. 667-678.

[23] L. Obrst, "Ontologies for semantically interoperable systems" Proceedings of the twelfth international conference on Information and knowledge management, 2003, pp. 366-369.

[24] H.-K. Lin, J. A. Harding, and M. Shahbaz, "Manufacturing system engineering ontology for semantic interoperability across extended project teams," International journal of production research, vol 42, no. 24, 2004, pp. 5099-5118.

[25] I. Robinson, J. Webber, and E. Eifrem, "Graph Databases: New Opportunities for Connected Data," O'Reilly Media Inc., 2015.

[26] T. J. Lampoltshammer, and S. Wiegand, "Improving the Computational Performance of Ontology-Based Classification Using Graph Databases" Remote Sensing, vol. 7, no. 7, 2015, pp. 9473-9491.

[27] J. J. Miller, "Graph database applications and concepts with Neo4j," Proceedings of the Southern Association for Information Systems Conference, Atlanta, vol. 2324, 2013, pp. 141-147.

[28] Non-profit Risk Management Centre (NRMC), "Fact Sheet: Accident/Incident/Near Miss Investigation," http://www.nonprofitrisk.org/tools/workplace-safety/public-sector/concepts/acc-in-nm-ps.htm, 2016, [last visited 20-12-2016].

[29] Blacket, C, "Combining accident analysis techniques for organizational safety," Ph.D. Thesis. School of Computer Science and Informatics National University of Ireland. 2015.

[30] R. Battle, and D. Kolas, "Geosparql: enabling a geospatial semantic web," Semantic Web Journal, vol. 3, no. 4, 2011, pp. 355-370.

Short Papers and Student Contributions

Improving Maintenance Processes with Data Science

How Machine Learning Opens Up New Possibilities

Dorian Prill, Simon Kranzer, Robert Merz
Department Information Technology & Systems Management
Salzburg University of Applied Sciences
Salzburg, Austria

***Abstract*—**In this presentation we briefly describe potential benefits of using data analysis methods to improve maintenance processes. After a short introduction to an automated, multi-step maintenance process and a survey of the state of data in industry, we explain, how selected data analysis methods can be used to improve maintenance demand detection

***Keywords*—**Predictive maintenance, failure detection, machine learning, industrial data analysis

I. A FULLY INTEGRATED MAINTENANCE CYCLE

Maintenance has become a highly complex process to ensure the optimal performance, availability and reliability of every piece of equipment. Costly downtimes have to be avoided or minimized. Optimal scheduling during planned downtime, while avoiding unnecessary replacement of still functional components, is a key factor for reducing maintenance cost. Recently, fully automated, multi-step processes (Fig. 1) have been developed, which include the detection and classification of maintenance demand, selecting and planning appropriate maintenance procedures, scheduling and dispatching service personnel, providing instructions and guidance during maintenance procedures and collecting feedback to improve detection algorithms and maintenance plans [1].

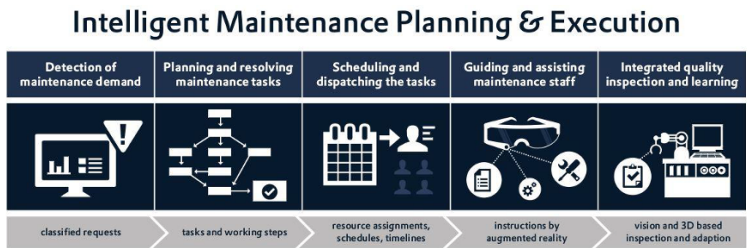


Fig. 1: Modern, automated maintenance process proposed by [1]

One of the most crucial and complicated steps in such a maintenance process is the first one; the correct detection and classification of maintenance demand. While the type of failure and the required countermeasures are quite obvious once the problem has already occurred, unforeseen breakdown of equipment causes downtime at undesirable moments, potential collateral damage, loss of reputation and huge financial cost. Therefore, it is required to reliably predict upcoming failures, their type and the remaining running life as accurately as possible.

In general, three approaches are used to detect abnormal behaviour:

A. Isolation of defective components

Measurement values and data are used to detect faulty behaviour of individual components of the machine or whether a trend towards faulty behaviour is apparent. E.g., knocking noises or vibrations indicate worn out bearings.

A. Isolation of causes of defect

Measurement values and data are used to detect conditions, that will ultimately be the trigger for faulty behaviour. E.g., high temperatures will cause lubricants to evaporate, which will lead to the breakage of an engine part.

B. Isolation of environmental conditions

Changes in environmental conditions play an important role in when and how frequently failures occur. By monitoring changes in environmental conditions predictions on remaining running life of components can be made. To develop and execute accurate and reliable algorithms for failure prediction, carefully designed data collection and data classification are required.

P. Haber et al. (Hrsg.), *Data Science – Analytics and Applications*, http://doi.org/10.1007/978-3-658-19287-7_15

II. THE STATE OF DATA IN INDUSTRY

The limited public exposure of the production industry, especially on the supplier side, presents a bias to the normal observer as data for internet usage, housing markets or even satellite images is comparatively easy to come by where production data is of no public concern. The reason for this is rather obvious, as sensitive data may be exploited to potentially harm a competitor's business, actively or passively.

Let us review what the state of data looks like and how it is expected to develop over time. The authors of [2], a survey from 2013, state an estimated annual volume of around 2 Exabyte combined for the manufacturing industry. One year before, in a paper [*3*] by General Electric on the use of their intelligent automation platform, a sample frequency of 5000 samples every 33 ms is reported. These numbers could already be considered "outdated" by today's digital standards when looking at the growth projections that have been made and came true, especially with the hype that the Internet-Of-Things (IOT) has experienced in recent years.

In 2015 an analysis [*4*] reported projected growth on the order of 101 to 102 in magnitude. IDC [6] in late 2016 predicted 180 Zettabytes of annually worldwide created data in 2025 and still 44 Zettabytes in 2020. According to IDC [6] the IoT data that is analyzed and used to transform business processes in 2025 will be 44 Zettabytes and machines will use real time data to make real time decisions. There will be 30 billion connected devices in 2020 and 80 billion in 2025 with 152,200 new connected devices per minute. There is no indicator how much of that data and devices are from industry or manufacturing [6]. Also General Electrics in 2017 published some numbers in their report on the rise of industrial Big Data [7]. One of their customers produces 4 trillion data samples per year.

Further, industrial data often lacks a verbose history of error occurrences, which is quite natural considering the overall goal of maximizing the operational time of a machine. This however, somewhat opposes the idea of a machine learning model where you want to infer unseen behaviour from past experience. Building a statistical model from a very small number of samples that outperforms an educated guess for a specific label ranges from challenging to impossible. Another, very practical problem, is the maturity of software infrastructure in industry. Software cycles are much longer than in other areas of business, as it usually involves multiple hardware platforms and has to conform to specific security regulations. Also the proven codebase is usually harder to maintain and change as it involves more low-level languages.

As the term 'Data Analytics' becomes more and more popular and accessible through available education and tools, manufacturers are realizing the potential this holds for their business. This is not only true for maintenance applications but can also be applied to e.g. product research, where one may find that customers are using your product in slightly different ways than intended by the manufacturer and different operating conditions may reveal design flaws or opportunity for expansion of the portfolio.

III. DETECTION OF MAINTENANCE DEMAND

A classical way of assessing the health status of equipment is to monitor the measured output and check for thresholds. This poses several problems though; The specific thresholds are often not known beforehand because they may vary with operating condition, and secondly, watching hard thresholds the condition may be recognized too late to schedule it in a fitting manner. Also, with the increasing number of parameters than can be measured in modern devices, the search for optimal thresholds is not practical. When working with larger amounts of data, we want to be able to make decision solely based on our observations (unless there is domain-knowledge available).

A. DATA-DRIVEN DECISIONS

This leads to the notion of data-driven decisions. The monitoring of processes should rely on the information present in the actual data if no meta-information is known. A tried and tested method for doing so is a family of techniques called "Control Charts" which have been conceived in the 1920's by Walter A. Shewhart who certainly was a key figure in the invention of statistical process control [8].

The idea is rather simple: Monitor the statistical parameters of a process and, if a small sequence of sample deviates from its mean observed value by more than a certain amount (usually 3-sigma) consider the observed state an irregular condition due to the low probability of this event under normal circumstances. This method works really well and is still widely used in industrial control software today. The question is: Given newly developed techniques, can we do better?

The advantage of using Machine Learning algorithms for this task is manifold. In real world scenarios it is usually the case that given data is not suitable as input to algorithms as-is, but has to merged from different sources, and features have to be engineered for the specific question that needs answering. Also contrary to the well behaved data sets you will find on well-known repositories [9], labels, or more specifically target variables, are usually not or only limitedly available, due to lack of maturity of software infrastructure.

This is the first issue that Machine Learning (ML) can help with. In this phase of knowledge discovery, unsupervised techniques can be used to identify potential anomalous behaviour. An example for such a simple yet powerful technique to find said instances is called "Isolation Forest" [8], which essentially assumes that outlier data is more likely to produce shorter path lengths when building a decision tree. From experience, this works surprisingly well if one has no or little knowledge about the desired target variable, which actually is a common problem as outlined before.

be used to improve over regular supervised techniques, although this is not always guaranteed and must be evaluated for each specific case.

After these stages, an expert's domain knowledge is used to analyze the found subset of points and assign the correct label

to it. This process is very labour intensive and the number of points should be reduced as much as possible.

The above applies to classification scenarios. Machine learning can also be used for time-series forecasting in which case the target variable is created from the following values of the input variable. If there is at least some knowledge about the target variable available, special semi-supervised techniques can be used to improve the decision boundaries.

IV. Outlook

Our current work is involved with implementing and proving practical solutions for predicting and optimally scheduling maintenance tasks and accurately estimating remaining running life until maintenance is required in the field of industrial automation. In an ongoing project involving historical data from large combustion engines, we are using a combination of techniques from signal processing and machine learning to develop a predictive model to identify early indicators of erroneous behavior. While investigations are still ongoing, early results show that deploying this model will allow for a more accurate planning of maintenance operations by increasing the time from detection to failure, while at the same time providing an estimate for the remaining running time.

References

[1] Kranzer, S., et. al., An Intelligent Maintenance Planning Framework Prototype for Production Systems, IEEE ICIT 2017, 18th Annual International Conference on Industrial Technology, Toronto, March 22-25, 2017

[2] *M. Bailly and J. Manyika, ''Is manufacturing 'cool' again?'' McKinsey Global Inst., Jan. 21, 2013.*

[3] *General Electric Intelligent Platforms, ''The rise of industrial big data,'' 2012, White Paper.*

[4] Shen Yin, Okyay Kaynak, Big Data for Modern Industry: Challenges and Trends, Proceedings of the IEEE, Vol 103, No. 2, February 2015

[5] Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation Forest." *2008 Eighth IEEE International Conference on Data Mining* (2008): n. pag.Web.https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf

[6] Press, Gil, IoT Mid-Year Update From IDC And Other Research Firms, 2016, https://www.forbes.com/sites/gilpress/2016/08/05/iot-mid-year-update-from-idc-and-other-research-firms/#6613b3db55c5

[7] General Electrics, The Rise of Industrial Big Data, 2017, http://leadwise.mediadroit.com/files/19174the_rise_of_industrial_big_data_wp_gft834.pdf

[8] Shewart, W A (1931). Economic Control of Quality of Manufactured Product. Van Nordstrom. p. 18.

[9] Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

ouRframe

A Graphical Workflow Tool for R

Marco Gruber, Elisabeth Birnbacher and Tobias Fellner
Department Information Technology & Systems Management
Salzburg University of Applied Sciences
Salzburg, Austria

***Abstract*—*ouRframe* is a graphical workflow tool, which is based on the statistical programming language R, and which allows its users to create individual, mathematical data analysis models by assembling function blocks in a drawing area. It can be used for all data analyses that can be performed in R and it can be extended with customized blocks. For now, ouRframe is implemented as a prototype, but in the future, it could be further developed into a fully functional Open Source product.**

Keywords*—*R; Data Analysis; Graphical Workflow Tool

I. Introduction

Analyzing data is a complex and challenging task. Even though the idea of collecting data and extracting information from it appears to be simple, it can be quite difficult to achieve the desired results, since the data analysis process is highly iterative and non-linear [1]. Tools, which are created for the examination and interpretation of data sets, cannot change the fact that there is no simple step-by-step approach for analyzing data, but they can support data scientists and enable them to explore it [2]. This paper describes such a supportive tool, which is called *ouRframe*. It's a framework, in its appearance similar to RapidMiner [3], that has been created by three students of the Information Technology and Systems Management Master's program of the Salzburg University of Applied Sciences in the course of a research and development project.

II. DESCRIPTION OF THE TOOL

ouRframe is a framework that allows its users to design and execute individual, mathematical data analysis models via an intuitive and easy-to-use graphical user interface.

A. R – The basis of ouRframe

For the required calculations, *ouRframe* uses the statistical programming language R [4], which was chosen as basis for the framework because it offers several advantages. Firstly, R is Open Source, which was important for the project team, since it wanted to keep the possibility open to also publish the framework as an Open Source project later on. Secondly, R provides a wide variety of various packages that can be used to individualize your programs. Especially, the different types of plots, which are implemented in R or that can be added to R programs with the help of packages, were of interest for the framework. They enable the user to visualize the data not just with standard plots, but also with specialized graphics [2]. This quality of R proves to be particularly helpful during exploratory data analyses, which usually precede actual data analyses [5]. Another benefit of R is that it has a large community, which is continuously working on its improvement and adding new features to it in form of packages.

B. Technical details

The framework has been implemented as a web-based application, so that several clients can use it simultaneously. The frontend has been developed in Angular 4, the backend was created in C#. For the integration of R, R.NET was used.

C. Graphical User Interface (GUI)

ouRframe follows a modular principle, which means that the user can assemble data analysis models from different building blocks, which the framework provides, by simply selecting different function blocks and connecting them with lines between them.

The following figure displays a section of the browser GUI, which the user can interact with. It can be divided in four main sectors.

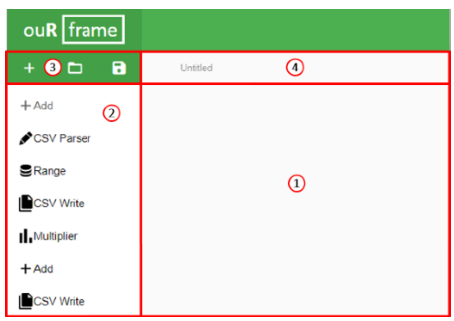


Fig. 1: Four sectors of the graphical user interface

P. Haber et al. (Hrsg.), *Data Science – Analytics and Applications*, http://doi.org/10.1007/978-3-658-19287-7_16

The first sector is the drawing area. Here users can create their own data analysis models by dragging and dropping function blocks from the menu on the left (sector two) into the area and by connecting them with lines. The connection lines can be drawn by clicking on the input of one block and the output of another one. In order to avoid erroneous models only blocks that are compatible with each other can be combined. Additionally, the blocks can be parameterized when the user selects it with a double click. In this case a menu opens, where the user can enter parameters that depend on the block, which was chosen. If the user clicks e.g. on a *kmeans* function block, a parameter could be the number of clusters, which the clustering algorithm has to find.

The menu in sector two displays all function blocks that the framework offers e.g. the *CSV Parser Block*, which can be used to read data from a *CSV* File into the framework or the *R Plot Block,* which enables users to display data sets, with any plot that R provides. The framework also gives the users the possibility to add their own, customized blocks with the option *Add Block*. When the user clicks on the respective button, an entry mask is open, where the user can enter the name of the new block, the icon, which is used to display it, the names and data types of the block inputs, outputs and parameters and the R code, which determines the functionality of the block.

Above the block menu, the user can find the operations menu (sector three), which enables the user to store and load models. It also offers the option to open an empty drawing area in a new tab. The user can switch between different models by selecting the respective tab in sector four.

III. APPLICATION EXAMPLE

ouRframe can be used for all data analyses that can be performed in R. The following figure displays a simple example and shows how the model for a linear regression looks like in the drawing area.

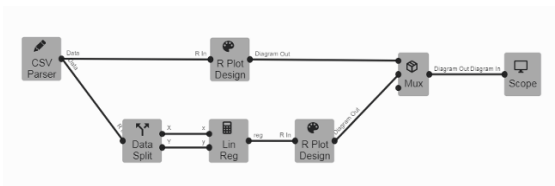


Fig. 2: Model for a linear regression

On the far left, the *CSV Parser* can be seen, which reads the data from a *CSV* file and provides it to the other blocks in the form of a matrix. The data is then used by two different blocks. The *R Plot Design* block prepares the data so that it can be displayed in the *Scope* block. By double clicking on this block, the user has the possibility to select which columns of the data set should be shown in the scope, which color it should have and whether the data points shall be displayed as a scatter plot or as a continuous line. In this case the original data is displayed as a scatter block. The *Data Split* block takes the matrix and splits it into two arrays that can be processed by the *Lin Reg* block, which is responsible for the linear regression. The results of this block are then handed to a second *R Plot Design* block that makes sure that the results are displayed as a red line. In the end, the *Scope* displays the results of the data analysis as shown in Fig. 3, when you click on the *Scope* block.

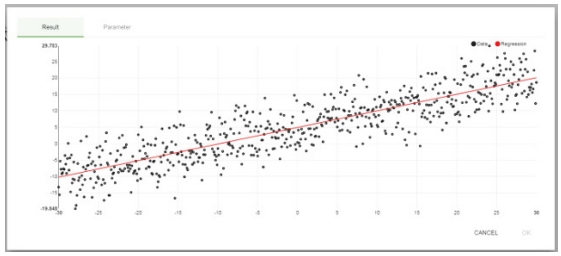


Fig. 3: Result of the linear regression example

IV. OUTLOOK

For now, the framework is implemented as a prototype that serves as a proof-of-concept for the idea of a graphical workflow tool for R. We plan to further develop the tool in the context of future research projects at the Salzburg University of Applied Sciences. It is further intended to publish ouRframe as an Open Source project in the future.

ACKNOWLEDGMENT

We would like to thank our project supervisors FH-Prof. DI Dr. Robert Merz and DI (FH) DI Simon Kranzer for their great support throughout the entire research and development project as well as the organizers of the 1st International Data Science Conference for choosing us as speakers at the Students Talks of the conference. Our special thanks goes to Maximilian Tschuchnig, Astrid Karnutsch and FH-Ass.Prof. DI (FH) DI Peter Haber for their help during the preparation of our presentation.

REFERENCES

[1] E. Matsui and R. Peng, The Art of Data Science - A Guide for Anyone Who Works with Data, Leanpub, 2015.

[2] J. Chambers, W. Härdle and D. Hand, Software for Data Analysis - Programming with R, New York: Springer, 2008.

[3] "RapidMiner," [Online]. Available: https://rapidminer.com/. [Accessed 30 June 2017].

[4] "The R Project for Statistical Computing," [Online]. Available: https://www.r-project.org/. [Accessed 29 June 2017].

[5] R. Schutt and C. O'Neil, Doing Data Science, Köln u.a.: O'Reilly, 2013.

Sentiment Analysis

A Students Point of View

Hofer Dominik
Informationstechnologie und Systemmanagement
Salzburg University of Applied Sciences
Salzburg, Austria

Abstract— **Sentiment Analysis (SA) is a new, fast growing scientific field, which makes it quite difficult for people, e.g.: marketing executives, sociologists, etc. to stay up to date to the vast possibilities, this field offers. But also for students, who are interested in learning a subject, apart from university, this task can be quite demanding. Due to technological advancements, it is easy to gain knowledge about aspects of SA, but it still takes time to experiment and analyze various techniques. Therefore, in this presentation, there will be an overview of the different approaches of SA, and how some of them can be applied. This includes the coding language Python, libraries/toolkits, and the involvement of social media. The primary goal is to give an overview of existing possibilities of SA implementations.**

Keywords—sentiment analysis; probability; level; toolkit; libraries;

I. Introduction

Due to technological advancement, intelligent systems can be used in everyday lives. Especially when it comes to speech recognition products such as "Alexa" have claimed a lot of attention. But this Amazon product is merely the tip of the ice berg and creating a simple system that allows someone to analyze spoken or written words is no longer difficult. The knowledge to create these systems can be gained online for free. The problem rather is the vast amount of knowledge and losing the overview of what source to use first.

II. Definition of Sentiment Analysis

Before someone can create something, it is necessary to know what someone intends to create. Definitions can be rather confusing but are necessary to gain a clear picture of what someone intends to create. In the case of sentiment analysis, there are a couple of definitions which all refer to sentiment analysis as a synonym for opinion mining, emotion AI, or natural language processing [1] [2].
But it needs to be said that sentiment does not stop when it comes to words. A person in its whole behavior shows sentiment. This includes the way a person talks, the body language, as well as the use of emoticons. According to this thought, sentiment analysis could be applied to every aspect of a person's behavior.

III. Basic Theory

It is not only enough to understand the concept of one's future creation, the basic principles also need to be understood. In the case of sentiment analysis these concepts would mean, polarity, level of usage, and the different approaches, someone can choose from.

A. Polarity

- There are two types of polarity. The first one would be "normal" polarity, which represents the states of positive, neutral, and negative [1]. It is also possible to create five states, which would add very positive and very negative to the already given three. The second option would be beyond polarity [1]. In this case a system tries to figure out the actual emotional state of a person: happy, surprised, peaceful, etc.

B. Level of Usage

- The level of usage refers to the ways a system has o approach an analysis. The first level would be the document level, where an entire document is analyzed [3]. In this case the resulting sentiment refers to the entire document. The second level is the sentence level. Contrary to the document level, the resulting sentiments refer to single sentences [3]. The last level would be the aspect level [3]. There the aspect on a sentence gets analyzed. An example would be: "Although I love chocolate, this chocolate muffin is dreadful.". On a sentence level, the polarity would be around zero and therefore not giving any kind of information. On the aspect level, on the contrary, the polarity for chocolate would be positive and for the muffin, on the other hand, negative.

C. Approaches

- There are two main approaches, when it comes to sentiment analysis. The first one would be lexicon-based, and as the name already implies, this approach can be further divided into a dictionary-based, and an corpus-based approach [4]. On the other hand, there is the machine learning approach, which can be divided into supervised and unsupervised learning [4]. An

P. Haber et al. (Hrsg.), *Data Science – Analytics and Applications*, http://doi.org/10.1007/978-3-658-19287-7_17

(semi) unsupervised approach is not recommended, due to its low score [5]. Supervised learning is further divided into 4 categories; probabilistic classifiers being the focus group.

IV. Probabilistic Classifiers

There are three classifiers that are widely used.

A. *Naïve Bayes*

- NB assumes that the effect of a predict (x) on a given class (c) is independent of the values of other predictors [6]. This principle can be explained with the analyzation of an apple. If the object is round, red colored, and has a diameter of 10 cm, it is an apple. If not, it is not

B. *Bayesian Network*

- It is a graphical model, which represents a set of random variables and their dependencies [7]. Using the joint probability allows to calculate different events.

C. *Maximum Entropy*

- This classifier is a form of a-priori probability and aims at figuring out the one data set with the most entropy, which will then be further used.

These classifiers are examples. The usefulness of a classifier varies due to the aim of the analyzation and the quality and amount of data.

V. Toolkit and Libraries

The basic difference between a toolkit and libraries is, that the libraries in a toolkit are guaranteed to work together. But using different libraries, which are not in a toolkit might not work together, which may cause unforeseen difficulties. When it comes to toolkits, the natural language toolkit (NLTK) is one of the best known. It includes over 50 corpora and lexical resources, text processing libraries and example programs [8]. For users who are more interested in using a library, TextBlob is a possibility. It is a python library focusing on processing textual data, and has example programs [9].

VI. Alternatives for non-coders

There are possibilities for people, who are not interested in coding, to do sentiment analysis. The first option would be Weka. It is a collection of machine learning algorithms for data tasks. Created by the University of Waikato and named after the bird, that only lives there. Videos about how to use WEKA can be found on the main page or on Youtube. The second possibility would be the Data Scientist Workbench by IBM. It is an online platform dedicated to give data scientist the possibility to work online and with different tools such as: Spark, SystemT, Seahorse, etc. [11]. The main benefit of this site is, that a user, who attends online courses at the IBM Big Data University can apply the knowledge, which was taught in a class. Some courses even use the Data Scientist Workbench for laboratory purposes.

VII. Conclusion

Sentiment analysis is a mighty tool, which allows machines to analyze natural language. But before analyzing language, it is necessary to understand what sentiment analysis means and how it can be applied. Furthermore, it is worth to mention, that there are already vast possibilities of analyzing languages and even for people, who have no knowledge of coding, do platforms exist, where the analyzations will be done for them. It should also not be forgotten, that sentiment analysis does not only apply to spoken or written words, but could also be applied to the entire way a person communicates. All in all, sentiment analysis is a very diverse field and still promises vast opportunities.

References

[1] J. Pustejovsky, A. Stubbs, “The Basics,” Natural Language Annotation for Machine Learning,, first edition. Sebastopol: O’Reilly, 2013, Chap. 1, pp. 1 – 31.

[2] Lexalytics. (2017, June 30). Sentiment Analysis: What is Sentiment Analysis [Online]. Website, Access: https://www.lexalytics.com/technology/sentiment

[3] B. Liu, “The Problem of Sentiment Analysis,” Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, 1st edition. New York: Cambridge University Press, 2015, Chap 2, pp. 17 – 45.

[4] Sciencedirect. (2017, June 30). Sentiment Analysis Algorithms and Applications: A Survey [Online]. Website, Access: http://www.sciencedirect.com/science/article/pii /S2090447914000550

[5] Nadeau, D., Turney, P.D., and Matwin, S. (2006), Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity, Proceedings of the 19th Canadian Conference on Artificial Intelligence (CAI-06), Quebec City, Canada, pp. 266-277.

[6] Saedsayad. (2017, June 30). Naïve Bayesian [Online]. Website, Access: http://www.saedsayad.com/naive_bayesian.htm

[7] Rationalistramble (2017, June 30). Bayesian Networks [Online]. Websie, Access: https://rationalistramble.wordpress.com/2015/12/09/bayesian-networks/

[8] NLTK (2017, June 30). Natural Language Toolkit [Online]. Website, Access: www.nltk.org.

[9] Textblob (2017, 30 June). TextBlob: Simplified Text Processing [Online]. Website, Access: textblob.readthedocs.io/en/dev/index.html.

[10] WEKA University of Waikato (2017, June 30). Weka 3: Data Mining Software in Java [Online]. Website, Access: http://www.cs.waikato.ac.nz/ml/weka/index.html

[11] IBM Data Scientist Workbench (2017, June 30). Data Scientist Workbench [Online]. Access: https://datascientistworkbench.com