

Studies in Computational Intelligence 943

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

Complex Networks & Their Applications IX

Volume 1, Proceedings of the Ninth
International Conference on Complex
Networks and Their Applications
COMPLEX NETWORKS 2020

 Springer

Studies in Computational Intelligence

Volume 943

Series Editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland

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

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

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

More information about this series at <http://www.springer.com/series/7092>

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

Complex Networks & Their Applications IX

Volume 1, Proceedings of the Ninth
International Conference on Complex
Networks and Their Applications
COMPLEX NETWORKS 2020

 Springer

Editors

Rosa M. Benito
Grupo de Sistemas Complejos
Universidad Politécnica de Madrid
Madrid, Madrid, Spain

Chantal Cherifi
IUT Lumière
University of Lyon
Bron Cedex, France

Hocine Cherifi
LIB, UFR Sciences et Techniques
Université de Bourgogne
Dijon, France

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

Luis Mateus Rocha
Center for Social and Biomedical
Complexity, Luddy School of Informatics,
Computing, and Engineering
Indiana University
Bloomington, IN, USA

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

ISSN 1860-949X

ISSN 1860-9503 (electronic)

Studies in Computational Intelligence

ISBN 978-3-030-65346-0

ISBN 978-3-030-65347-7 (eBook)

<https://doi.org/10.1007/978-3-030-65347-7>

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2021

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

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

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

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

Preface

This 2020 edition of the International Conference on Complex Networks & Their Applications is the ninth of a series that began in 2011. Over the years, this adventure has made the conference one of the major international events in network science.

Network science continues to trigger a tremendous interest among the scientific community of various fields such as finance and economy, medicine and neuroscience, biology and earth sciences, sociology and politics, computer science and physics. The variety of scientific topics ranges from network theory, network models, network geometry, community structure, network analysis and measure, link analysis and ranking, resilience and control, machine learning and networks, dynamics on/of networks, diffusion and epidemics, visualization. It is also worth mentioning some recent applications with high added value for current trend social concerns such as social and urban networks, human behavior, urban systems—mobility, or quantifying success. The conference brings together researchers that study the world through the lens of networks. Catalyzing the efforts of this scientific community, it drives network science to generate cross-fertilization between fundamental issues and innovative applications, review the current state of the field and promote future research directions.

Every year, researchers from all over the world gather in our host venue. This year's edition was initially to be hosted in Spain by Universidad Politécnica de Madrid. Unfortunately, the COVID-19 global health crisis forced us to organize the conference as a fully online event.

Nevertheless, this edition attracted numerous authors with 323 submissions from 51 countries. The papers selected for the volumes of proceedings clearly reflect the multiple aspects of complex network issues as well as the high quality of the contributions.

All the submissions were peer-reviewed by 3 independent reviewers from our strong international program committee. This ensured high-quality contributions as well as compliance to conference topics. After the review process, 112 papers were selected to be included in the proceedings.

Undoubtedly, the success of this edition relied on the authors who have produced high-quality papers, as well as the impressive list of keynote speakers who delivered fascinating plenary lectures:

- Leman Akoglu (Carnegie Mellon University, USA): “Graph-Based Anomaly Detection: Problems, Algorithms and Applications”
- Stefano Boccaletti (Florence University, Italy): “Synchronization in Complex Networks, Hypergraphs and Simplicial Complexes”
- Fosca Giannotti (KDD Lab, Pisa, Italy): “Explainable Machine Learning for Trustworthy AI”
- János Kertész (Central European University, Hungary): “Possibilities and Limitations of using mobile phone data in exploring human behavior”
- Vito Latora (Queen Mary University of London, UK): “Simplicial model of social contagion”
- Alex “Sandy” Pentland (MIT Media Lab, USA): “Human and Optimal Networked Decision Making in Long-Tailed and Non-stationary Environments”
- Nataša Pržulj (Barcelona Supercomputing Center, Spain): “Untangling biological complexity: From omics network data to new biomedical knowledge and Data-Integrated Medicine”

The topics addressed in the keynote talks allowed a broad coverage of the issues encountered in complex networks and their applications to complex systems.

For the traditional tutorial sessions prior to the conference, our two invited speakers delivered insightful talks. David Garcia (Complexity Science Hub Vienna, Austria) gave a lecture entitled “Analyzing complex social phenomena through social media data,” and Mikko Kivela (Aalto University, Finland) delivered a talk on “Multilayer Networks.”

Each edition of the conference represents a challenge that cannot be successfully achieved without the deep involvement of many people, institutions and sponsors.

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

We record our thanks to our fellow members of the Organizing Committee. José Fernando Mendes (University of Aveiro, Portugal), Jesús Gomez Gardeñes (University of Zaragoza, Spain) and Huijuan Wang (TU Delft, Netherlands) chaired the lightning sessions. Manuel Marques-Pita (Universidade Lusófona, Portugal), José Javier Ramasco (IFISC, Spain) and Taha Yasseri (University of Oxford, UK) managed the poster sessions. Luca Maria Aiello (Nokia Bell Labs, UK) and Leto Peel (Université Catholique de Louvain, Belgium) were our tutorial chairs. Finally, Sabrina Gaito (University of Milan, Italy) and Javier Galeano (Universidad Politécnica de Madrid, Spain) were our satellite chairs.

We extend our thanks to Benjamin Renoust (Osaka University, Japan), Michael Schaub (MIT, USA), Andreia Sofia Teixeira (Indiana University Bloomington, USA), Xiangjie Kong (Dalian University of Technology, China), the publicity

chairs for advertising the conference in America, Asia and Europa, hence encouraging the participation.

We would like also to acknowledge Regino Criado (Universidad Rey Juan Carlos, Spain) as well as Roberto Interdonato (CIRAD - UMR TETIS, Montpellier, France) our sponsor chairs.

Our deep thanks go to Matteo Zignani (University of Milan, Italy), publication chair, for the tremendous work he has done at managing the submission system and the proceedings publication process.

Thanks to Stephany Rajeh (University of Burgundy, France), Web chair, in maintaining the Web site.

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

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

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

Generally, we are thankful to all those who have helped us contributing to the success of this meeting. Sincere thanks to the contributors, and the success of the technical program would not be possible without their creativity.

Finally, we would like to express our most sincere thanks to the program committee members for their huge efforts in producing high-quality reviews in a very limited time.

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

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

Organization and Committees

General Chairs

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

Advisory Board

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

Program Chairs

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

Satellite Chairs

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

Lightning Chairs

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

Poster Chairs

Manuel Marques-Pita
José Javier Ramasco
Taha Yasseri

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

Publicity Chairs

Benjamin Renoust
Andreia Sofia Teixeira
Michael Schaub
Xiangjie Kong

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

Tutorial Chairs

Luca Maria Aiello
Leto Peel

Nokia Bell Labs, UK
UCLouvain, Belgium

Sponsor Chairs

Roberto Interdonato
Regino Criado

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

Local Committee Chair

Juan Carlos Losada

Universidad Politécnica de Madrid, Spain

Local Committee

Jacobo Aguirre
David Camacho
Julia Martinez-Atienza
Juan Manuel Pastor
Leticia Perez Sienes
Francisco Prieto
Fabio Revuelta

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

Publication Chair

Matteo Zignani

University of Milan, Italy

Web Chair

Stephany Rajeh

University of Burgundy, France

Program Committee

Jacobo Aguirre	Centro Nacional de Biotecnología, Spain
Amreen Ahmad	Jamia Millia Islamia, India
Masaki Aida	Tokyo Metropolitan University, Japan
Luca Maria Aiello	Nokia Bell Labs, UK
Marco Aiello	University of Stuttgart, Germany
Esra Akbas	Oklahoma State University, USA
Mehmet Aktas	University of Central Oklahoma, USA
Tatsuya Akutsu	Kyoto University, Japan
Reka Albert	The Pennsylvania State University, USA
Aleksandra Aloric	Institute of Physics Belgrade, Serbia
Claudio Altafini	Linköping University, Sweden
Benjamin Althouse	New Mexico State University, USA
Lucila G. Alvarez-Zuzek	IFIMAR-UNMDP, Argentina
Luiz G. A. Alves	Northwestern University, USA
Enrico Amico	Swiss Federal Institute of Technology in Lausanne, Switzerland
Hamed Amini	Georgia State University, USA
Chuankai An	Dartmouth College, USA
Marco Tulio Angulo	National Autonomous University of Mexico (UNAM), Mexico
Demetris Antoniadis	RISE Research Center, Cyprus
Alberto Antonioni	Carlos III University of Madrid, Spain
Nino Antulov-Fantulin	ETH Zurich, Switzerland
Nuno Araujo	Universidade de Lisboa, Portugal
Elsa Arcaute	University College London, UK
Laura Arditti	Polytechnic of Turin, Italy
Samin Aref	Max Planck Institute for Demographic Research, Germany
Panos Argyrakis	Aristotle University of Thessaloniki, Greece
Malbor Asllani	University of Limerick, Ireland
Tomaso Aste	University College London, UK
Martin Atzmueller	Tilburg University, Netherlands
Konstantin Avrachenkov	Inria, France
Jean-Francois Baffier	National Institute of Informatics, Japan
Giacomo Baggio	University of Padova, Italy
Rodolfo Baggio	Bocconi University, Italy
Franco Bagnoli	University of Florence, Italy
Annalisa Barla	Università di Genova, Italy
Paolo Barucca	University College London, UK
Anastasia Baryshnikova	Calico Life Sciences, USA
Nikita Basov	St. Petersburg State University, Russia
Gareth Baxter	University of Aveiro, Portugal

Marya Bazzi	University of Oxford, UK
Mariano Beguerisse Diaz	University of Oxford, UK
Andras A. Benczur	Hungarian Academy of Sciences, Hungary
Rosa M. Benito	Universidad Politécnica de Madrid, Spain
Luis Bettencourt	University of Chicago, USA
Ginestra Bianconi	Queen Mary University of London, UK
Ofer Biham	The Hebrew University of Jerusalem, Israel
Livio Bioglio	University of Turin, Italy
Hanjo Boekhout	Leiden University, Netherlands
Johan Bollen	Indiana University Bloomington, USA
Christian Bongiorno	Università degli Studi di Palermo, Italy
Anton Borg	Blekinge Institute of Technology, Sweden
Stefan Bornholdt	Universität Bremen, Germany
Federico Botta	The University of Warwick, UK
Alexandre Bovet	Université Catholique de Louvain, Belgium
Dan Braha	NECSI, USA
Ulrik Brandes	ETH Zürich, Switzerland
Markus Brede	University of Southampton, UK
Marco Bressan	Sapienza University of Rome, Italy
Piotr Bródka	Wroclaw University of Science and Technology, Poland
Javier M. Buldu	Universidad Rey Juan Carlos, Spain
Raffaella Burioni	Università di Parma, Italy
Fabio Caccioli	University College London, UK
Rajmonda Caceres	Massachusetts Institute of Technology, USA
Carmela Calabrese	University of Naples Federico II, Italy
Paolo Campana	University of Cambridge, UK
M. Abdullah Canbaz	Indiana University Kokomo, USA
Carlo Vittorio Cannistraci	TU Dresden, Germany
Vincenza Carchiolo	Università di Catania, Italy
Giona Casiraghi	ETH Zurich, Switzerland
Douglas Castilho	Federal University of Minas Gerais, Brazil
Costanza Catalano	Gran Sasso Science Institute, Belgium
Remy Cazabet	Lyon University, France
David Chavalarias	CNRS, CAMS/ISC-PIF, France
Kwang-Cheng Chen	University of South Florida, USA
Po-An Chen	National Chiao Tung University, Taiwan
Xihui Chen	University of Luxembourg, Luxembourg
Xueqi Cheng	Institute of Computing Technology, China
Chantal Cherifi	Lyon 2 University, France
Hocine Cherifi	University of Burgundy, France
Peter Chin	Boston University, USA
Matteo Chinazzi	Northeastern University, USA
Matteo Cinelli	University of Rome “Tor Vergata”, Italy
Richard Clegg	Queen Mary University of London, UK

Reuven Cohen	Bar-Ilan University, Israel
Alessio Conte	University of Pisa, Italy
Marco Coraggio	University of Naples Federico II, Italy
Michele Coscia	IT University of Copenhagen, Denmark
Clementine Cottineau	CNRS, Centre Maurice Halbwachs, France
Regino Criado	Universidad Rey Juan Carlos, Spain
Mihai Cucuringu	University of Oxford and The Alan Turing Institute, USA
Marcelo Cunha	IFBA, Brazil
Giulio Valentino Dalla Riva	University of Canterbury, New Zealand
Kareem Darwish	Qatar Computing Research Institute, Qatar
Bhaskar Dasgupta	University of Illinois, Chicago, USA
Joern Davidsen	University of Calgary, Canada
Toby Davies	University College London, UK
Pasquale De Meo	Vrije Universiteit Amsterdam, Italy
Fabrizio De Vico Fallani	Inria - ICM, France
Charo I. del Genio	Coventry University, UK
Pietro Delellis	University of Naples Federico II, Italy
Jean-Charles Delvenne	University of Louvain, Belgium
Yong Deng	Xi'an Jiaotong University, China
Bruce Desmarais	The Pennsylvania State University, USA
Patrick Desrosiers	Université Laval, Canada
Riccardo Di Clemente	University of Exeter, UK
Matías Di Muro	Universidad Nacional de Mar del Plata-CONICET, Argentina
Jana Diesner	University of Illinois at Urbana-Champaign, USA
Shichang Ding	University of Goettingen, Germany
Linda Douw	Amsterdam UMC, Netherlands
Johan Dubbeldam	Delft University of Technology, Netherlands
Jordi Duch	Universitat Rovira i Virgili, Spain
Kathrin Eismann	University of Bamberg, Germany
Mohammed El Hassouni	Mohammed V University in Rabat, Morocco
Andrew Elliott	University of Oxford, UK
Michael T. M. Emmerich	Leiden University, Netherlands
Frank Emmert-Streib	Tampere University of Technology, Finland
Gunes Ercal	SIUE, USA
Alexandre Evsukoff	COPPE/UFRJ, Brazil
Mauro Faccin	Université Catholique de Louvain, Belgium
Sofia Fernandes	Laboratory of Artificial Intelligence and Decision Support, Portugal
Guilherme Ferraz de Arruda	ISI Foundation, Italy
Daniel Figueiredo	COPPE/UFRJ, Brazil
Jorge Finke	Pontificia Universidad Javeriana, Colombia
Marco Fiore	IMDEA Networks Institute, Spain
Alessandro Flammini	Indiana University Bloomington, USA

Manuel Foerster	Bielefeld University, Germany
Barbara Franci	Delft University of Technology, Netherlands
Diego Função	University of São Paulo, Brazil
Angelo Furno	University of Lyon and University Gustave Eiffel, France
Sabrina Gaito	University of Milan, Italy
Lazaros Gallos	Rutgers University, USA
José Manuel Galán	Universidad de Burgos, Spain
Joao Gama	University of Porto, Portugal
Yerali Gandica	Université Catholique de Louvain, Belgium
Jianxi Gao	Rensselaer Polytechnic Institute, USA
David Garcia	Medical University of Vienna and Complexity Science Hub Vienna, Austria
Federica Garin	Inria, France
Michael Gastner	Yale-NUS College, Singapore
Alexander Gates	Northeastern University, USA
Vincent Gauthier	Telecom SudParis/Institut Mines-Telecom, France
Raji Ghawi	Technical University of Munich, Germany
Tommaso Gili	IMT School for Advanced Studies Lucca, Italy
Silvia Giordano	SUPSI, Switzerland
Rosalba Giugno	University of Verona, Italy
David Gleich	Purdue University, USA
Antonia Godoy	Rovira i Virgili University, Spain
Kwang-Il Goh	Korea University, South Korea
Jaime Gomez	Universidad Politécnica de Madrid, Spain
Jesus Gomez-Gardenes	Universidad de Zaragoza, Spain
Antonio Gonzalez	Universidad Autónoma de Madrid, Spain
Bruno Gonçalves	New York University, USA
Joana Gonçalves-Sá	Nova School of Business and Economics, Portugal
Przemyslaw Grabowicz	University of Massachusetts, Amherst, USA
Carlos Gracia-Lázaro	BIFI, Spain
Justin Gross	UMass Amherst, USA
Jelena Grujic	Vrije Universiteit Brussel, Belgium
Jean-Loup Guillaume	L3i - Université de la Rochelle, France
Mehmet Gunes	Stevens Institute of Technology, USA
Sergio Gómez	Universitat Rovira i Virgili, Spain
Meesoon Ha	Chosun University, South Korea
Jürgen Hackl	University of Liverpool, Switzerland
Edwin Hancock	University of York, UK
Chris Hankin	Imperial College London, UK
Jin-Kao Hao	University of Angers, France
Heather Harrington	University of Oxford, UK

Yukio Hayashi	Japan Advanced Institute of Science and Technology, Japan
Mark Heimann	University of Michigan, USA
Torsten Heinrich	University of Oxford, Germany
Denis Helic	Graz University of Technology, Austria
Chittaranjan Hens	Indian Institute of Chemical Biology, India
Laura Hernandez	Université de Cergy-Pontoise, France
Samuel Heroy	University of Oxford, UK
Takayuki Hiraoka	Aalto University, Finland
Philipp Hoevel	University College Cork, Ireland
Petter Holme	Tokyo Institute of Technology, Japan
Seok-Hee Hong	University of Sydney, Australia
Ulrich Hoppe	University of Duisburg-Essen, Germany
Yanqing Hu	Sun Yat-sen University, China
Flavio Iannelli	Humboldt University, Germany
Yuichi Ikeda	Kyoto University, Japan
Roberto Interdonato	CIRAD - UMR TETIS, France
Giulia Iori	City, University of London, UK
Antonio Iovanella	University of Rome “Tor Vergata”, Italy
Gerardo Iñiguez	Central European University, Hungary
Sarika Jalan	IIT Indore, India
Mahdi Jalili	RMIT University, Australia
Jaroslawn Jankowski	West Pomeranian University of Technology, Poland
Marco Alberto Javarone	Coventry University, UK
Hawoong Jeong	Korea Advanced Institute of Science and Technology, South Korea
Tao Jia	Southwest University, China
Chunheng Jiang	Rensselaer Polytechnic Institute, USA
Ming Jiang	University of Illinois at Urbana-Champaign, USA
Di Jin	Tianjin University, China
Di Jin	University of Michigan, USA
Ivan Jokić	Delft University of Technology, Netherlands
Bertrand Jouve	CNRS, France
Jason Jung	Chung-Ang University, South Korea
Marko Jusup	Tokyo Institute of Technology, Japan
Arkadiusz Jędrzejewski	Wrocław University of Science and Technology, Poland
Byungham Kahng	Seoul National University, South Korea
Rushed Kanawati	Université Paris 13, France
Rowland Kao	University of Edinburgh, UK
Márton Karsai	ENS de Lyon, France
Eytan Katzav	The Hebrew University of Jerusalem, Israel
Mehmet Kaya	Firat University, Turkey

Domokos Kelen	Institute for Computer Science and Control, Hungary
Dror Kenett	Johns Hopkins University, USA
Yoed Kenett	University of Pennsylvania, USA
Janos Kertesz	Central European University, Hungary
Mohammad Khansari	University of Tehran, Iran
Hamamache Kheddouci	Universite Claude Bernard, France
Hyoungshick Kim	Sungkyunkwan University, South Korea
Jinseok Kim	University of Michigan, USA
Maksim Kitsak	Northeastern University, USA
Mikko Kivela	Aalto University, Finland
Konstantin Klemm	IFISC (CSIC-UIB), Spain
Peter Klimek	Medical University of Vienna, Austria
Dániel Kondor	SMART, Singapore
Xiangjie Kong	Dalian University of Technology, China
Ismo Koponen	University of Helsinki, Finland
Onerva Korhonen	Université de Lille, Finland
Jan Kralj	Jozef Stefan Institute, Slovenia
Reimer Kuehn	King's College London, UK
Prosenjit Kundu	National Institute of Technology Durgapur, India
Ryszard Kutner	University of Warsaw, Poland
Haewoon Kwak	Qatar Computing Research Institute, Qatar
Richard La	University of Maryland, USA
Hemank Lamba	Carnegie Mellon University, USA
Renaud Lambiotte	University of Oxford, UK
Aniello Lampo	UOC, Spain
Christine Largeron	Université de Lyon, France
Jennifer Larson	New York University, USA
Anna T. Lawniczak	University of Guelph, Ontario, Canada
Eric Leclercq	University of Burgundy, France
Deok-Sun Lee	Inha University, South Korea
Sune Lehmann	Technical University of Denmark, Denmark
Balazs Lengyel	Hungarian Academy of Sciences, Hungary
Juergen Lerner	University of Konstanz, Germany
Fabrizio Lillo	University of Bologna, Italy
Ji Liu	Stony Brook University, USA
Yang-Yu Liu	Harvard University, USA
Giacomo Livan	University College London, UK
Lorenzo Livi	University of Manitoba, Canada
Alessandro Longheu	University of Catania, Italy
Laura Lotero	Universidad Pontificia Bolivariana, Colombia
Meilian Lu	Beijing University of Posts and Telecommunications, China
John C. S. Lui	The Chinese University of Hong Kong, Hong Kong

Leonardo Maccari	Ca' Foscari University of Venice, Italy
Matteo Magnani	Uppsala University, Sweden
Cécile Mailler	UVSQ, France
Nishant Malik	Rochester Institute of Technology, USA
Fragkiskos Malliaros	University of Paris-Saclay, France
Noel Malod-Dognin	University College London, UK
Giuseppe Mangioni	University of Catania, Italy
Ed Manley	University of Leeds, UK
Rosario Nunzio Mantegna	Palermo University, Italy
Madhav Marathe	University of Virginia, USA
Manuel Sebastian Mariani	University of Zurich, Switzerland
Radek Marik	Czech Technical University, Czechia
Andrea Marino	University of Florence, Italy
Antonio Marques	Universidad Rey Juan Carlos, Spain
Manuel Marques-Pita	Universidade Lusofona, Portugal
Christoph Martin	Leuphana University of Lüneburg, Germany
Cristina Masoller	Universitat Politècnica de Catalunya, Spain
Emanuele Massaro	Ecole Polytechnique Fédérale de Lausanne, Switzerland
Rossana Mastrandrea	IMT Institute of Advanced Studies, Italy
John Matta	SIUE, USA
Arya McCarthy	Johns Hopkins University, USA
Fintan Mcgee	Gabriel Lippmann Public Research Centre, Ireland
Matúš Medo	University of Electronic Science and Technology of China, China
Jörg Menche	CeMM of the Austrian Academy of Sciences, Austria
Jose Fernando Mendes	University of Aveiro, Portugal
Ronaldo Menezes	University of Exeter, UK
Humphrey Mensah	Syracuse University, USA
Anke Meyer-Baese	FSU, USA
Radosław Michalski	Wrocław University of Science and Technology, Poland
Tijana Milenkovic	University of Notre Dame, USA
Letizia Milli	University of Pisa, Italy
Andreea Minca	Cornell University, USA
Shubhanshu Mishra	University of Illinois at Urbana-Champaign, USA
Bivas Mitra	Indian Institute of Technology Kharagpur, India
Marija Mitrovic	Institute of physics Belgrade, Serbia
Andrzej Mizera	University of Luxembourg, Luxembourg
Osnat Mokryn	University of Haifa, Israel
Roland Molontay	Budapest University of Technology and Economics, Hungary
Raul Mondragon	Queen Mary University of London, UK

Misael Mongiovi	Consiglio Nazionale delle Ricerche, Italy
Andres Moreira	Universidad Tecnica Federico Santa Maria, Chile
Paolo Moretti	Friedrich-Alexander-University Erlangen-Nurnberg, Germany
Esteban Moro	Universidad Carlos III de Madrid, Spain
Greg Morrison	University of Houston, USA
Sotiris Moschoyiannis	University of Surrey, UK
Elisha Moses	The Weizmann Institute of Science, Israel
Igor Mozetič	Jozef Stefan Institute, Slovenia
Animesh Mukherjee	Indian Institute of Technology, India
Masayuki Murata	Osaka University, Japan
Tsuyoshi Murata	Tokyo Institute of Technology, Japan
Alessandro Muscoloni	TU Dresden, Germany
Matthieu Nadini	New York University, Italy
Zachary Neal	Michigan State University, USA
Muaz Niazi	COMSATS Institute of IT, Pakistan
Rolf Niedermeier	TU Berlin, Germany
Peter Niemeyer	Leuphana Universität Lüneburg, Germany
Jordi Nin	Universitat Ramon Llull, Spain
Rogier Noldus	Ericsson, Netherlands
El Faouzi Nour-Eddin	IFSTTAR, France
Neave O'Clery	University College London, UK
Masaki Ogura	Nara Institute of Science and Technology, Japan
Marcos Oliveira	Leibniz Institute for the Social Sciences, USA
Andrea Omicini	Università degli Studi di Bologna, Italy
Luis Ospina-Forero	University of Manchester, UK
Gergely Palla	Statistical and Biological Physics Research Group of HAS, Hungary
Pietro Panzarasa	Queen Mary University of London, UK
Fragkiskos Papadopoulos	Cyprus University of Technology, Cyprus
Symeon Papadopoulos	Information Technologies Institute, Greece
Michela Papandrea	SUPSI, Switzerland
Francesca Parise	MIT, USA
Han Woo Park	Yeungnam University, South Korea
Juyong Park	KAIST, South Korea
Fabio Pasqualetti	UC Riverside, USA
Leto Peel	Universite Catholique de Louvain, Belgium
Tiago Peixoto	Central European University and ISI Foundation, Germany
Matjaz Perc	University of Maribor, Slovenia
Hernane Pereira	UEFS and SENAI CIMATEC, Brazil
Lilia Perfeito	Nova SBE, Portugal
Chiara Perillo	University of Zurich, Switzerland
Giovanni Petri	ISI Foundation, Italy
Jürgen Pfeffer	Technical University of Munich, Germany

Carlo Piccardi	Politecnico di Milano, Italy
Flavio Pinheiro	Universidade NOVA de Lisboa, USA
Clara Pizzuti	CNR-ICAR, Italy
Chiara Poletto	Sorbonne University, France
Maurizio Porfiri	New York University Tandon School of Engineering, USA
Pawel Pralat	Ryerson University, Canada
Victor Preciado	University of Pennsylvania, USA
Natasza Przulj	University College London, Spain
Oriol Pujol	University of Barcelona, Spain
Rami Puzis	Ben Gurion University of the Negev, Israel
Christian Quadri	University of Milan, Italy
Marco Quaggiotto	ISI Foundation, Italy
Filippo Radicchi	Northwestern University, USA
Tomasz Raducha	Faculty of Physics, University of Warsaw, Poland
Jose J. Ramasco	IFISC (CSIC-UIB), Spain
Felix Reed-Tsochas	University of Oxford, UK
Gesine Reinert	University of Oxford, UK
Benjamin Renoust	Osaka University, Japan
Daniel Rhoads	Universitat Oberta de Catalunya, Spain
Pedro Ribeiro	University of Porto, Portugal
Massimo Riccaboni	IMT Institute for Advanced Studies Lucca, Italy
Laura Ricci	University of Pisa, Italy
Alessandro Rizzo	Politecnico di Torino, Italy
Celine Robardet	INSA Lyon, France
Luis E. C. Rocha	Ghent University, Belgium
Luis M. Rocha	Indiana University Bloomington, USA
Francisco Rodrigues	University of São Paulo, Brazil
Fernando Rosas	Imperial College London, UK
Giulio Rossetti	KDD Lab ISTI-CNR, Italy
Camille Roth	CNRS, Germany
Celine Rozenblat	University of Lausanne, Institut de Géographie, Switzerland
Giancarlo Ruffo	Università di Torino, Italy
Meead Saberi	UNSW, Australia
Ali Safari	Friedrich-Alexander Universität Erlangen-Nürnberg, Germany
Marta Sales-Pardo	Universitat Rovira i Virgili, Spain
Arnaud Sallaberry	Université Paul Valéry Montpellier 3, France
Iraj Saniee	Bell Labs, Alcatel-Lucent, USA
Francisco C. Santos	Universidade de Lisboa, Portugal
Jari Saramäki	Aalto University, Finland
Koya Sato	University of Tsukuba, Japan
Hiroki Sayama	Binghamton University, USA
Antonio Scala	Italian National Research Council, Italy

Michael Schaub	University of Oxford, UK
Maximilian Schich	The University of Texas at Dallas, USA
Frank Schweitzer	ETH Zurich, Switzerland
Santiago Segarra	Rice University, USA
Irene Sendiña-Nadal	Rey Juan Carlos University, Spain
M. Ángeles Serrano	Universitat de Barcelona, Spain
Saray Shai	Wesleyan University, USA
Aneesh Sharma	Google, USA
Rajesh Sharma	University of Tartu, Estonia
Julian Sienkiewicz	Warsaw University of Technology, Poland
Anurag Singh	NIT Delhi, India
Lisa Singh	Georgetown University, USA
Rishabh Singhal	Dayalbagh Educational Institute, India
Sudeshna Sinha	Indian Institutes of Science Education and Research, India
Per Sebastian Skardal	Trinity College, USA
Oskar Skibski	University of Warsaw, Poland
Michael Small	The University of Western Australia, Australia
Keith Smith	University of Edinburgh, UK
Igor Smolyarenko	Brunel University London, UK
Zbigniew Smoreda	Orange Labs, France
Tom Snijders	University of Groningen, Netherlands
Annalisa Socievole	National Research Council of Italy, Italy
Igor M. Sokolov	Humboldt University of Berlin, Germany
Albert Sole	Universitat Rovira i Virgili, Spain
Sucheta Soundarajan	Syracuse University, USA
Jaya Sreevalsan-Nair	IIIT Bangalore, India
Massimo Stella	Institute for Complex Systems Simulation, UK
Arkadiusz Stopczynski	Technical University of Denmark, Denmark
Blair D. Sullivan	University of Utah, USA
Xiaoqian Sun	Beihang University, China
Xiaoqian Sun	Chinese Academy of Sciences, China
Pål Sundsøy	NBIM, Norway
Samir Suweis	University of Padua, Italy
Boleslaw Szymanski	Rensselaer Polytechnic Institute, USA
Bosiljka Tadic	Jozef Stefan Institute, Slovenia
Andrea Tagarelli	DIMES, University of Calabria, Italy
Kazuhiro Takemoto	Kyushu Institute of Technology, Japan
Frank Takes	Leiden University and University of Amsterdam, Netherlands
Fabien Tarissan	ENS Paris-Saclay (ISP), France
Dane Taylor	University at Buffalo, SUNY, USA
Claudio Juan Tessone	Universität Zürich, Switzerland
François Théberge	Tutte Institute for Mathematics and Computing, Canada

Olivier Togni	Burgundy University, France
Ljiljana Trajkovic	Simon Fraser University, Canada
Jan Treur	Vrije Universiteit Amsterdam, Netherlands
Milena Tsvetkova	London School of Economics and Political Science, UK
Liubov Tupikina	Ecole Polytechnique, France
Janos Török	Budapest University of Technology and Economics, Hungary
Stephen Uzzo	New York Hall of Science, USA
Lucas D. Valdez	FAMAF-UNC, Argentina
Pim van der Hoorn	Eindhoven University of Technology, Netherlands
Piet Van Mieghem	Delft University of Technology, Netherlands
Michalis Vazirgiannis	AUEB, Greece
Balazs Vedres	University of Oxford, UK
Wouter Vermeer	Northwestern University, USA
Christian Lyngby Vestergaard	CNRS and Institut Pasteur, France
Anil Kumar Vullikanti	University of Virginia, USA
Johannes Wachs	Central European University, Hungary
Huijuan Wang	Delft University of Technology, Netherlands
Lei Wang	Beihang University, China
Ingmar Weber	Qatar Computing Research Institute, Qatar
Guanghai Wen	Southeast University, China
Gordon Wilfong	Bell Labs, USA
Mateusz Wilinski	Scuola Normale Superiore di Pisa, Italy
Richard Wilson	University of York, UK
Dirk Witthaut	Forschungszentrum Jülich, Germany
Bin Wu	Beijing University of Posts and Telecommunications, China
Jinshan Wu	Beijing Normal University, China
Feng Xia	Federation University Australia, Australia
Haoxiang Xia	Dalian University of Technology, China
Xiaoke Xu	Dalian Minzu University, China
Gitanjali Yadav	University of Cambridge, UK
Gang Yan	Tongji University, China
Xiaoran Yan	Indiana University Bloomington, USA
Taha Yasseri	University of Oxford, UK
Ying Ye	Nanjing University, China
Qingpeng Zhang	City University of Hong Kong, USA
Zi-Ke Zhang	Hangzhou Normal University, China
Junfei Zhao	Columbia University, USA
Matteo Zignani	University of Milan, Italy
Eugenio Zimeo	University of Sannio, Italy
Lorenzo Zino	University of Groningen, Netherlands
Antonio Zippo	Consiglio Nazionale delle Ricerche, Italy

Fabiana Zollo
Arkaitz Zubiaga
Claudia Zucca

Ca' Foscari University of Venice, Italy
Queen Mary University of London, UK
University of Glasgow, UK

Contents

Community Structure

A Method for Community Detection in Networks with Mixed Scale Features at Its Nodes	3
Soroosh Shalileh and Boris Mirkin	
Efficient Community Detection by Exploiting Structural Properties of Real-World User-Item Graphs	15
Larry Yueli Zhang and Peter Marbach	
Measuring Proximity in Attributed Networks for Community Detection	27
Rinat Aynulin and Pavel Chebotarev	
Core Method for Community Detection	38
A. A. Chepovskiy, S. P. Khaykova, and D. A. Leshchev	
Effects of Community Structure in Social Networks on Speed of Information Diffusion	51
Nako Tsuda and Sho Tsugawa	
Closure Coefficient in Complex Directed Networks	62
Mingshan Jia, Bogdan Gabrys, and Katarzyna Musial	
Nondiagonal Mixture of Dirichlet Network Distributions for Analyzing a Stock Ownership Network	75
Wenning Zhang , Ryohei Hisano, Takaaki Ohnishi, and Takayuki Mizuno	
Spectral Clustering for Directed Networks	87
William R. Palmer and Tian Zheng	
Composite Modularity and Parameter Tuning in the Weight-Based Fusion Model for Community Detection in Node-Attributed Social Networks	100
Petr Chunaev, Timofey Gradov, and Klavdiya Bochenina	

Maximal Labeled-Cliques for Structural-Functional Communities	112
Debajyoti Bera	
Community Detection in a Multi-layer Network Over Social Media	124
Maham Mobin Sheikh and Rauf Ahmed Shams Malick	
Using Preference Intensity for Detecting Network Communities	137
József Dombi and Sakshi Dhama	
Community Detection Algorithm Using Hypergraph Modularity	152
Bogumił Kamiński, Paweł Prałat, and François Théberge	
Towards Causal Explanations of Community Detection in Networks	164
Georgia Baltsoy, Anastasios Gounaris, Apostolos N. Papadopoulos, and Konstantinos Tsihclas	
A Pledged Community? Using Community Detection to Analyze Autocratic Cooperation in UN Co-sponsorship Networks	177
Cosima Meyer and Dennis Hammerschmidt	
Distances on a Graph	189
Pierre Miasnikof, Alexander Y. Shestopaloff, Leonidas Pitsoulis, Alexander Ponomarenko, and Yuri Lawryshyn	
Local Community Detection Algorithm with Self-defining Source Nodes	200
Saharnaz Dilmaghani, Matthias R. Brust, Gregoire Danoy, and Pascal Bouvry	
Investigating Centrality Measures in Social Networks with Community Structure	211
Stephany Rajeh, Marinette Savonnet, Eric Leclercq, and Hocine Cherifi	
Network Analysis	
Complex Network Analysis of North American Institutions of Higher Education on Twitter	225
Dmitry Zinoviev, Shana Cote, and Robert Díaz	
Connectivity-Based Spectral Sampling for Big Complex Network Visualization	237
Jingming Hu, Seok-Hee Hong, Jialu Chen, Marnijati Torkel, Peter Eades, and Kwan-Liu Ma	
Graph Signal Processing on Complex Networks for Structural Health Monitoring	249
Stefan Bloemheugel, Jurgen van den Hoogen, and Martin Atzmueller	
An Analysis of Four Academic Department Collaboration Networks with Respect to Gender	262
Lauren Nakamichi, Theresa Migler, and Zoë Wood	

Uncovering the Image Structure of Japanese TV Commercials Through a Co-occurrence Network Representation 273
 Mariko I. Ito and Takaaki Ohnishi

Movie Script Similarity Using Multilayer Network Portrait Divergence 284
 Majda Lafhel, Hocine Cherifi, Benjamin Renoust, Mohammed El Hassouni, and Youssef Mourchid

Interaction of Structure and Information on Tor 296
 Mahdieh Zabihimayvan, Reza Sadeghi, Dipesh Kadariya, and Derek Doran

Classifying Sleeping Beauties and Princes Using Citation Rarity 308
 Takahiro Miura, Kimitaka Asatani, and Ichiro Sakata

Finding High-Degree Vertices with Inclusive Random Sampling 319
 Yitzchak Novick and Amotz BarNoy

Concept-Centered Comparison of Semantic Networks 330
 Darkhan Medeuov, Camille Roth, Ksenia Puzyreva, and Nikita Basov

Diffusion and Epidemics

Analyzing the Impact of Geo-Spatial Organization of Real-World Communities on Epidemic Spreading Dynamics 345
 Alexandru Topîrceanu

Identifying Biomarkers for Important Nodes in Networks of Sexual and Drug Activity 357
 Jacob Grubb, Derek Lopez, Bhuvaneshwar Mohan, and John Matta

Opinion Dynamic Modeling of Fake News Perception 370
 Cecilia Toccaceli, Letizia Milli, and Giulio Rossetti

Influence Maximization for Dynamic Allocation in Voter Dynamics . . . 382
 Zhongqi Cai, Markus Brede, and Enrico Gerding

Effect of Interaction Mechanisms on Facebook Dynamics Using a Common Knowledge Model 395
 Chris J. Kuhlman, Gizem Korkmaz, S. S. Ravi, and Fernando Vega-Redondo

Using Link Clustering to Detect Influential Spreaders 408
 Simon Krukowski and Tobias Hecking

Prediction of the Effects of Epidemic Spreading with Graph Neural Networks 420
 Sebastian Mežnar, Nada Lavrač, and Blaž Škrlj

Learning Vaccine Allocation from Simulations	432
Gerrit Großmann, Michael Backenköhler, Jonas Klesen, and Verena Wolf	
Suppressing Epidemic Spreading via Contact Blocking in Temporal Networks	444
Xunyi Zhao and Huijuan Wang	
Blocking the Propagation of Two Simultaneous Contagions over Networks	455
Henry L. Carscadden, Chris J. Kuhlman, Madhav V. Marathe, S. S. Ravi, and Daniel J. Rosenkrantz	
Stimulation Index of Cascading Transmission in Information Diffusion over Social Networks	469
Kazufumi Inafuku, Takayasu Fushimi, and Tetsuji Satoh	
Diffusion Dynamics Prediction on Networks Using Sub-graph Motif Distribution	482
Alexey L. Zaykov, Danila A. Vaganov, and Valentina Y. Guleva	
Using Distributed Risk Maps by Consensus as a Complement to Contact Tracing Apps	494
Miguel Rebollo, Rosa M. Benito, Juan C. Losada, and Javier Galeano	
Dynamics on/of Networks	
Distributed Algorithm for Link Removal in Directed Networks	509
Azwirman Gusrialdi	
Data Compression to Choose a Proper Dynamic Network Representation	522
Remy Cazabet	
Effect of Nonisochronicity on the Chimera States in Coupled Nonlinear Oscillators	533
K. Premalatha, V. K. Chandrasekar, M. Senthilvelan, R. Amuda, and M. Lakshmanan	
Evolution of Similar Configurations in Graph Dynamical Systems	544
Joshua D. Priest, Madhav V. Marathe, S. S. Ravi, Daniel J. Rosenkrantz, and Richard E. Stearns	
Congestion Due to Random Walk Routing	556
Onuttom Narayan, Iraj Saniee, and Vladimir Marbukh	
Strongly Connected Components in Stream Graphs: Computation and Experimentations	568
Léo Rannou, Clémence Magnien, and Matthieu Latapy	

The Effect of Cryptocurrency Price on a Blockchain-Based Social Network 581
 Cheick Tidiane Ba, Matteo Zignani, Sabrina Gaito, and Gian Paolo Rossi

Multivariate Information in Random Boolean Networks 593
 Sebastián Orellana and Andrés Moreira

Earth Sciences Applications

Complexity of the Vegetation-Climate System Through Data Analysis 609
 Andrés F. Almeida-Ñañay, Rosa M. Benito, Miguel Quemada, Juan C. Losada, and Ana M. Tarquis

Towards Understanding Complex Interactions of Normalized Difference Vegetation Index Measurements Network and Precipitation Gauges of Cereal Growth System 620
 David Rivas-Tabares and Ana M. Tarquis

Spatio-Temporal Clustering of Earthquakes Based on Average Magnitudes 627
 Yuki Yamagishi, Kazumi Saito, Kazuro Hirahara, and Naonori Ueda

Information Spreading in Social Media

Analyzing the Robustness of a Comprehensive Trust-Based Model for Online Social Networks Against Privacy Attacks 641
 Nadav Voloch, Ehud Gudes, and Nurit Gal-Oz

Media Partisanship During Election: Indonesian Cases 651
 Ardian Maulana and Hokky Situngkir

Media Polarization on Twitter During 2019 Indonesian Election 660
 Ardian Maulana and Hokky Situngkir

Influence of Retweeting on the Behaviors of Social Networking Service Users 671
 Yizhou Yan, Fujio Toriumi, and Toshiharu Sugawara

Author Index 683

Community Structure



A Method for Community Detection in Networks with Mixed Scale Features at Its Nodes

Soroosh Shalileh^(✉) and Boris Mirkin

Department of Data Science and Artificial Intelligence, NRU HSE Moscow
Russian Federation, 11, Pokrovski Boulevard, 109028 Moscow, Russia
sr.shalileh@gmail.com, bmirkin@hse.ru
<https://cs.hse.ru/>

Abstract. The problem of community detection in a network with features at its nodes takes into account both the graph structure and node features. The goal is to find relatively dense groups of interconnected entities sharing some features in common. Algorithms based on probabilistic community models require the node features to be categorical. We use a data-driven model by combining the least-squares data recovery criteria for both, the graph structure and node features. This allows us to take into account both quantitative and categorical features. After deriving an equivalent complementary criterion to optimize, we apply a greedy-wise algorithm for detecting communities in sequence. We experimentally show that our proposed method is effective on both real-world data and synthetic data. In the cases at which attributes are categorical, we compare our approach with state-of-the-art algorithms. Our algorithm appears competitive against them.

Keywords: Attributed network · Feature-rich network · Community detection · Mixed scale clustering · One by one clustering

1 Introduction: Previous Work and Motivation

Community detection is a popular field of data science with various applications ranging from sociology to biology to computer science. Recently this concept was extended from flat and weighted networks to networks with a feature space associated with its nodes. A community is a group, or cluster, of densely interconnected nodes that are similar in the feature space too. There have been published a number of papers proposing various approaches to identifying communities in feature-rich networks (see recent reviews in [8] and [3]). They naturally fall in three groups: (a) those heuristically transforming the feature-based data to augment the network format, (b) those heuristically converting the data to the features only format, and (c) those involving, usually, a probabilistic model of the phenomenon to apply the maximum likelihood principle for estimating its parameters. A typical method within approach (a) or (b) combines a number

of heuristical approaches, thus involving a number of unsubstantiated parameters which are rather difficult to put to a system, the more so to testing. Most interesting approaches in the modeling group (c) are represented by methods in [21] and [16]. The former statistically models inter-relation between the network structure and node attributes, the latter involves Bayesian inferences.

Our approach relates to that of modeling, except that we model the data rather than the process of data generation. Specifically, our data-driven model assumes a hidden partition of the node set in non-overlapping communities and parameters encoding the average within-community link intensity and feature central points. To find this partition and parameters, we apply a combined least-squares criterion to recover the data from the partition. We propose a greedy-wise procedure for finding clusters one-by-one, as already proved successful in application to both feature data only and network/similarity data only [2, 12]. In contrast to other approaches, this one is applicable to mixed scale data after categories are converted into 1/0 dummy variables considered as quantitative ones. Our experiments show that this approach is valid and competitive against state-of-the-art approaches.

The rest of the paper is organized as follows. We describe our model and algorithm in Sect. 2. In Sect. 3, we describe the setting of our experiments. In Sect. 4, we describe results of our experiments to validate our method and compare it with competition. We draw conclusions in Sect. 5.

2 A Least Squares Criterion

Let us consider a dataset represented by two matrices: a symmetric $N \times N$ network adjacency matrix $P = (p_{ij})$, where p_{ij} can be any reals, and by an $N \times V$ entity-to-feature matrix $Y = (y_{iv})$ with $i \in I$, I being an N -element entity set.

We assume that there is a partition $S = \{S_1, S_2, \dots, S_K\}$ of I in K non-overlapping communities, a.k.a. clusters, related to this dataset as described below.

Denote k -th cluster binary membership vector by $s_k = (s_{ik})$, $k = 1, 2, \dots, K$, so that its i -th component is equal to unity for $i \in S_k$, and zero otherwise. The cluster is assigned with a V -dimensional center vector $c_k = (c_{kv})$. Also, there is a positive network intensity weight of k -th cluster denoted by λ_k , to adjust the binary s_{ik} values to the measurement scale of the network adjacency matrix P .

Equations (1) and (2) below:

$$y_{iv} = \sum_{k=1}^K s_{ik} c_{kv} + f_{iv}, i \in I, v \in V, \quad (1)$$

$$p_{ij} = \sum_{k=1}^K \lambda_k s_{ik} s_{jk} + e_{ij}, i, j \in I. \quad (2)$$

express our model. Here values e_{ij} and f_{iv} are residuals that should be made as small as possible.

According to the least-squares principle, “right” membership vectors s_k , community centers c_k and intensity weights λ_k are minimizers of the summary least-squares criterion:

$$F(\lambda_k, s_k, c_k) = \rho \sum_{k=1}^K \sum_{i,v} (y_{iv} - c_{kv} s_{ik})^2 + \xi \sum_{k=1}^K \sum_{i,j} (p_{ij} - \lambda_k s_{ik} s_{jk})^2 \quad (3)$$

The factors ρ and ξ in Eq. (3) are expert-driven constants to balance the two sources of data.

On the first glance, criterion in Eq. (3) differs from what follows from Eqs. (2) and (1): the operation of summation over k is outside of the parentheses in it, whereas these equations require that to be within the parentheses. However, the formulation in (3) is consistent with the models in (2) and (1) because vectors s_k ($k = 1, 2, \dots, K$) correspond to a partition and thus are mutually orthogonal: For any specific $i \in I$, s_{ik} is zero for all k except one; that one k at which $i \in S_k$. Therefore, each of the sums over k in Eqs. (2) and (1) consists of just one item, so that the summation sign may be applied outside of the parentheses indeed.

To use a one-by-one clustering strategy [13] here, let us denote an individual community by S ; its center in feature space, by c ; and the corresponding intensity weight, by λ (just removing the index, k , for convenience). The extent of fit between the community and the dataset will be the corresponding part of criterion in (3):

$$F(\lambda, c_v, s_i) = \rho \sum_{i,v} (y_{iv} - c_v s_i)^2 + \xi \sum_{i,j} (p_{ij} - \lambda s_i s_j)^2 \quad (4)$$

The problem: given matrices $P = (p_{ij})$ and $Y = (y_{iv})$, find binary s , as well as real-valued λ and $c = (c_v)$, minimizing criterion (4).

As is well known, and, in fact, easy to prove, the optimal real-valued c_v is equal to the within- S mean of feature v , and the optimal intensity value λ is equal to the mean within-cluster link value:

$$c_v = \frac{\sum_{i \in S} y_{iv}}{|S|}; \quad \lambda = \frac{\sum_{i,j \in S} p_{ij}}{|S|^2} \quad (5)$$

Criterion (4) can be further reformulated as:

$$\begin{aligned} F(s) = & \rho \sum_{i,v} y_{iv}^2 - 2\rho \sum_{i,v} y_{iv} c_v s_i + \rho \sum_v c_v^2 \sum_i s_i^2 + \\ & \xi \sum_{i,j} p_{ij}^2 - 2\xi \lambda \sum_{i,j} p_{ij} s_i s_j + \xi \lambda^2 \sum_i s_i^2 \sum_j s_j^2 \end{aligned} \quad (6)$$

The items $T(Y) = \sum_{i,v} y_{iv}^2$ and $T(P) = \sum_{i,j} p_{ij}^2$ in (6) express quadratic scatters of data matrices Y and P , respectively. Using them, Eq. 6 can be reformulated as

$$F(s) = \rho T(Y) + \xi T(P) - G(s) \quad (7)$$

where

$$G(s) = 2\rho \sum_{i,v} y_{iv} c_v s_i - \rho \sum_v c_v^2 \sum_i s_i^2 + 2\xi\lambda \sum_{i,j} p_{ij} s_i s_j - \xi\lambda^2 \sum_i s_i^2 \sum_j s_j^2 \quad (8)$$

Equation (7) shows that the combined data scatter, $\rho T(Y) + \xi T(P)$ is decomposed in two complementary parts, one of which, $F(s)$, expresses the residual, that part of the data scatter which is minimized in Eqs. (1) and (2), whereas the other part, $G(s)$, expresses the contribution of the model to the data scatter.

By putting the optimal values c_v and λ from (5) into this expression, we obtain a simpler expression for $G(s)$

$$G = \rho |S| \sum_v c_v^2 + \xi\lambda \sum_{ij} p_{ij} s_i s_j \quad (9)$$

Maximizing G in (9) is equivalent to minimizing criterion F in 4 because of 7.

One can see that maximizing the first item in (9) requires obtaining a numerous cluster (the greater the $|S|$, the better) which is as far away from the space origin, 0, as possible (the greater the squared distance from 0, $|\sum_v c_v^2|$, the better). Usually the data are pre-processed so that the origin is shifted to the center of gravity, or grand mean, the point whose components are the averages of the corresponding features. In such a case, the goal of putting the cluster as far away from 0 as possible, means that the cluster should be anomalous. The second item in the criterion (9) is proportional to the sum of within-cluster links multiplied by the average within-cluster link λ . Maximizing criterion (9), thus, should produce a large anomalous cluster of a high density.

We employ a greedy heuristic: starting from arbitrary singleton $S = i$, the seed, add entities one by one so that the increment of G in (9) is maximized. After each addition, recompute optimal c_v and λ . Halt when the increment becomes negative. After stopping, the last check is executed: **Seed Relevance Check**: Remove the seed from the found cluster S . If the removal increases the cluster contribution; this seed is extracted from the cluster.

We refer to this algorithm as Feature-rich Network Addition Clustering algorithm, FNAC. Consecutive application of the algorithm FNAC to detect more than one community, forms our community detection algorithm SEFNAC below.

SEFNAC: Sequential Extraction of Feature-rich Network Addition Clusters

1. Initialization. Define $J = I$, the set of entities to which FNAC applies at every iteration, and set cluster counter $k = 1$.
2. Define matrices Y_J and P_J as parts of Y and P restricted at J . Apply FNAC at J , denote the output cluster S as S_k , its center c as c_k , the intensity λ as λ_k , and contribution G as G_k .

3. Redefine J by removing all the elements of S_k from it. Check whether thus obtained J is empty or not. If yes, stop. Define the current k as K and output all the solutions $S_k, c_k, \lambda_k, G_k, k = 1, 2, \dots, K$. If not, add 1 to k , and go to 2.

3 Setting of Experiments for Validation and Comparison of SEFNAC Algorithm

To set a computational experiment, one should specify its constituents:

1. The set of algorithms under comparison.
2. The set of datasets at which the algorithms are evaluated and/or compared.
3. The set of criteria for assessment of the experimental results.

3.1 Algorithms Under Comparison

We take two popular algorithms in the model-based approach, CESNA [21] and SIAN [16], which have been extensively tested in computational experiments. The author-made codes of the algorithms are publicly available in [11] and [14] respectively. We also tested the algorithm PAICAN from [1] in our experiments. The results of this algorithm, unfortunately, were always less than satisfactory; therefore, we exclude the algorithm PAICAN from this paper.

3.2 Datasets

We use both real world datasets and synthetic datasets.

Real World Datasets. We take on five real-world data sets listed in Table 1. Some of them involve both quantitative and categorical features. The algorithms under comparison, unlike the proposed algorithm SEFNAC, require that features are to be categorical. Therefore, whenever a data set contains a quantitative feature we convert that feature to a categorical version.

Malaria data set [9]

The nodes are amino acid sequences containing six highly variable regions (HVR) each. The edges are drawn between sequences with similar HVRs number 6. In this data set, there are two nominal attributes of nodes:

1. Cys labels derived from of a highly variable region HVR6 (assumed ground truth);
2. Cys-PoLV labels derived from the sequences adjacent to regions HVR 5 and 6.

Lawyers dataset [10]

The Lawyers dataset comes from a network study of corporate law partnership that was carried out in a Northeastern US corporate law firm, referred to as SG & R, 1988–1991, in New England. It is available for downloading at [19]. There is a friendship network between lawyers in the study. The features in this dataset are:

Table 1. Real world datasets under consideration. Symbols N, E, and F stand for the number of nodes, the number of edges, and the number of node features, respectively.

Name	Nodes	Edges	Features	Ground truth
Malaria HVR6 [9]	307	6526	6	Cys Labels
Lawyers [19]	71	339	18	Derived out of office and status features
World Trade [17]	80	1000	16	Derived out of continent and structural world system features
Parliament [1]	451	11646	108	Political parties
COSN [5]	46	552	16	Region

1. Status (partner, associate),
2. Gender (man, woman),
3. Office location (Boston, Hartford, Providence),
4. Years with the firm,
5. Age,
6. Practice (litigation, corporate),
7. Law school (Harvard or Yale, UCon., Other)

Most features are nominal. Two features, “Years with the firm” and “Age”, are quantitative. Authors of the previous studies converted them to the nominal format, accepted here too. The categories of “Years with the firm” are $x \leq 10$, $10 < x < 20$, and $x \geq 20$; the categories of “Age” are $x \leq 40$, $40 < x < 50$, and $x \geq 50$.

World-Trade dataset [17]

The World-Trade dataset contains data on trade between 80 countries in 1994. The link weights represent total imports by row-countries from column-countries, in \$1,000, for the class of commodities designated as ‘miscellaneous manufactures of metal’ to represent high technology products. The weights for imports with values less than 1% of the country’s total imports are zeroed. The node attributes are:

1. Continent (Africa, Asia, Europe, North America, Oceania, South America)
2. Structural World System Position (Core, Semi-Periphery, Periphery),
3. Gross Domestic Product per capita in \$ (GDP p/c)

We convert the GDP feature into a three-category nominal feature according to the minima of its histogram. The categories are: ‘Poor’ if GDP p/c is less than \$4406.9; ‘Mid-Range’ if GDP is between \$4406.9 and \$21574.5; and ‘Wealthy’ if GDP is greater than \$21574.5.

Parliament dataset[1]

The nodes correspond to members of the French Parliament. An edge is drawn if the corresponding MPs sign a bill together. The features are the constituency of MPs and their political party.

Consulting Organisational Social Network (COSN) dataset [5]

Nodes in this network correspond to employees in a consulting company. The (asymmetric) edges are formed in accordance with their replies to this question: “Please indicate how often you have turned to this person for information or advice on work-related topics in the past three months”. The answers are coded by 0 (I Do Not Know This Person), 1 (Never), 2 (Seldom), 3 (Sometimes), 4 (Often), and 5 (Very Often). These 6 numerals are the weights of the corresponding edges. Nodes in this network have the following attributes:

1. Organisational level (Research Assistant, Junior Consultant, Senior Consultant, Managing Consultant, Partner),
2. Gender (Male, Female),
3. Region (Europe, USA),
4. Location (Boston, London, Paris, Rome, Madrid, Oslo, Copenhagen).

Before applying SEFNAC, all attribute categories are converted into 1/0 dummy variables which are considered quantitative.

Generating Synthetic Data Sets. First of all, we specify the number of nodes N , the number of features V , and the number of communities, K , in a dataset to be generated. As the number of parameters to control is rather high, we narrow down the variation of our data generator by maintaining two types of settings only, a small size network and a medium size network. For a small size setting, we specify the values of the three parameters as follows: $N = 200$, $V = 5$, and $K = 5$. For the medium size, $N = 1000$, $V = 10$, and $K = 15$.

Generating Networks

At given numbers of nodes, N , and communities K , cardinalities of communities are defined uniformly randomly, up to a constraint that no community may have less than a pre-specified number of nodes (in our experiments, this is set to 30, so that probabilistic approaches are applicable), and the total number of nodes in all the communities sums to N .

Given the community sizes, we populate them with nodes, that are specified just by indices. Then we specify two probability values, p and q . Every within-community edge is drawn with the probability p , independently of other edges. Similarly, any between- community edge is drawn independently with the probability q .

Generating Quantitative Features

To model quantitative features, we generate each cluster from a Gaussian distribution whose covariance matrix is diagonal with diagonal values uniformly random in the range $[0.05, 0.1]$ to specify the cluster’s spread. Each component of the cluster center is generated uniformly random from the range $\alpha[-1, +1]$, so that the real positive α controls the cluster intermix: the smaller the α , the closer are cluster centers to each other.

In addition to cluster intermix, we take into account the possibility of presence of noise in data. We uniformly random generate a noise feature from an

interval defined by the maximum and minimum values. In this way, we may replicate 50% of the original data with noise features.

Generating Categorical Features

To model categorical features, we randomly choose the number of categories for each of them from the set $\{2, 3, \dots, L\}$ where $L = 10$ for small-size networks and $L = 15$ for the medium-size networks. Then, given the number of communities, K , and the numbers of entities, N_k for $(k = 1, \dots, K)$; the cluster centers are generated randomly so that no two centers may coincide at more than 50% of features.

Once a center of k -th cluster, $c_k = (c_{kv})$, is specified, N_k entities of this cluster are generated as follows. Given a pre-specified threshold of intermix, ϵ between 0 and 1, for every pair (i, v) , $i = 1 : N_k$; $v = 1 : V$, a uniformly random real number r between 0 and 1 is generated. If $r > \epsilon$, the entry x_{iv} is set to be equal to c_{kv} ; otherwise, x_{iv} is taken randomly from the set of categories specified for feature v .

Consequently, all entities in cluster k -th coincide with its center, up to rare errors if ϵ is close to 1. The smaller the epsilon, the more diverse, and thus intermixed, would be the generated entities.

Generating mixed scale features

We divide the number of features in two approximately equal parts, one to consist of quantitative features, the other, of categorical features. Each part is filled in independently, as described above.

3.3 Evaluation Criteria

To evaluate the result of a community detection algorithm, we compare the found partition with that generated by using: 1) the customary Adjusted Rand Index (ARI) [6] and 2) the Normalized Mutual Information (NMI) [4].

4 Results of Computational Experiments

The goal of our experiments is to test validity of the SEFNAC algorithm over all types of feature-rich network datasets under consideration. In the cases at which features are categorical, the SEFNAC algorithm is to be compared with the popular algorithms SIAN and CESNA.

4.1 Parameters of the Generated Datasets

We set network parameters, the probability of a within-community edge, p , and that between communities, q , to take either of two values each, $p = 0.7, 0.9$ and $q = 0.3, 0.6$. In the cases at which all the features are categorical, we decrease q -values to $q = 0.2, 0.4$, because all the three algorithms fail at $q = 0.6$. Feature generation is controlled by an intermix parameter, α at quantitative features, and ϵ at categorical features. We take each of the intermix parameters to be either 0.7 or 0.9.

Table 2. Performance of SEFNAC on synthetic networks combining quantitative and categorical features for two different sizes: The average ARI index and its standard deviation over 10 different data sets.

			Small-size networks			50% noisy feature			Medium-size networks			50% noisy features		
p	q	α/ϵ	ARI	NMI	K	ARI	NMI	K	ARI	NMI	K	ARI	NMI	K
0.9	0.3	0.9	0.99(0.01)	0.99(0.01)	5.00(0.00)	0.99(0.01)	0.99(0.01)	5.00(0.00)	1.00(0.00)	1.00(0.00)	15.00(0.00)	1.00(0.01)	1.00(0.01)	15.00(0.00)
0.9	0.3	0.7	0.98(0.03)	0.98(0.02)	5.00(0.00)	0.99(0.02)	0.99(0.02)	5.00(0.00)	1.00(0.00)	1.00(0.00)	15.00(0.00)	0.99(0.01)	0.99(0.01)	15.00(0.00)
0.9	0.6	0.9	0.91(0.01)	0.95(0.04)	4.60(0.50)	0.88(0.01)	0.92(0.05)	4.50(0.67)	0.95(0.08)	0.98(0.03)	14.00(1.26)	0.93(0.10)	0.97(0.04)	13.70(1.67)
0.9	0.6	0.7	0.86(0.14)	0.91(0.08)	4.80(0.60)	0.88(0.14)	0.91(0.09)	4.80(0.39)	0.84(0.08)	0.93(0.03)	12.10(1.22)	0.81(0.09)	0.92(0.04)	11.80(1.47)
0.7	0.3	0.9	0.99(0.02)	0.99(0.01)	5.00(0.00)	0.99(0.01)	0.99(0.01)	5.00(0.00)	0.99(0.01)	1.00(0.01)	14.90(0.30)	0.99(0.01)	1.00(0.01)	14.90(0.30)
0.7	0.3	0.7	0.94(0.10)	0.96(0.07)	4.90(0.30)	0.95(0.06)	0.96(0.04)	4.90(0.30)	0.99(0.01)	0.99(0.01)	14.80(0.40)	0.96(0.07)	0.98(0.03)	14.30(1.19)
0.7	0.6	0.9	0.74(0.20)	0.85(0.12)	3.80(0.87)	0.73(0.15)	0.83(0.10)	4.20(0.87)	0.56(0.14)	0.80(0.07)	7.80(1.78)	0.55(0.14)	0.80(0.07)	8.10(1.70)
0.7	0.6	0.7	0.67(0.14)	0.80(0.08)	4.30(1.10)	0.57(0.14)	0.73(0.10)	3.90(0.54)	0.39(0.09)	0.69(0.07)	7.10(1.51)	0.42(0.08)	0.71(0.05)	7.40(0.66)

To set a more realistic design, we may explicitly insert 50% features that are uniformly random in some datasets.

Therefore, generation of synthetic datasets is controlled by specifying six two-valued and one three-valued parameters: feature scales: quantitative, categorical, mixed; data size: small, medium; presence of noise features: yes, no; the probability of a within-community edge p ; the probability of a between-community edge q ; cluster inter-mix parameter α/ϵ . Therefore, there are 192 combinations of these altogether. At each setting, we generate 10 datasets, run a community detection algorithm, and calculate the mean and the standard deviation of ARI (NMI) values at these 10 datasets.

The following two sections present our experimental results for (a) testing validity of the SEFNAC algorithm at synthetic data, and (b) comparing performance of SEFNAC and competition on both real and synthetic data.

4.2 Validity of SEFNAC

Table 2 presents the results of our experiments at synthetic datasets with mixed scale features.

We can see that SEFNAC successfully recovers the numbers of communities at $q = 0.3$ and mostly fails at $q = 0.6$ – because this corresponds to a counter intuitive situation at which the probability of a link between separate communities is greater than 0.5. Yet even in this case the partition is recovered exactly when other parameters keep its structure tight, as say at $p = 0.9$. This holds for both small size and medium size cases. Insertion of noise features does reduce the levels of ARI (NMI) but not that much. The real reduction in the numbers of recovered communities, 7–8 out of 15 ones generated, occurs at the medium size data sets at really loose data structures with $p = 0.7$ and $q = 0.6$, leading to significant drops in the levels of ARI (NMI) values.

The picture is much similar at the cases of quantitative only and categorical only feature scales - we do not present them to shorten the paper.

4.3 Comparing SEFNAC and Competition

In this section, we compare the performance of SEFNAC with that of CESNA [21], and SIAN [16]. It should be reminded that SEFNAC determines the number of clusters automatically, whereas both CESNA and SIAN need that as part of the input.

Table 3 presents our results at synthetic datasets (with categorical features only, as required by the competition) and Table 4, at real world datasets.

Table 3. Comparison of CESNA, SIAN and SEFNAC at synthetic data sets with categorical features. The best results are highlighted using bold-face. The average ARI and NMI value and its standard deviation over 10 different data sets is reported.

			Small size networks						Medium size networks					
			CESNA		SIAN		SEFNAC		CESNA		SIAN		SEFNAC	
p	q	ϵ	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI
0.9	0.3	0.9	1.00(0.00)	1.00(0.00)	0.55(0.29)	0.58(0.30)	0.99(0.01)	0.99(0.01)	0.89(0.05)	0.94(0.03)	0.00(0.00)	0.00(0.00)	1.00(0.00)	1.00(0.00)
0.9	0.3	0.7	0.95(0.10)	0.97(0.06)	0.48(0.29)	0.52(0.27)	0.97(0.02)	0.97(0.03)	0.85(0.08)	0.92(0.03)	0.00(0.00)	0.00(0.00)	0.99(0.01)	0.99(0.01)
0.9	0.6	0.9	0.93(0.08)	0.93(0.06)	0.32(0.25)	0.37(0.27)	0.97(0.01)	0.96(0.02)	0.63(0.06)	0.75(0.04)	0.00(0.00)	0.00(0.00)	0.99(0.01)	1.00(0.00)
0.9	0.6	0.7	0.90(0.06)	0.90(0.06)	0.11(0.14)	0.12(0.15)	0.75(0.12)	0.73(0.11)	0.48(0.09)	0.66(0.06)	0.00(0.00)	0.00(0.00)	0.96(0.03)	0.97(0.01)
0.7	0.3	0.9	0.97(0.08)	0.97(0.04)	0.55(0.16)	0.60(0.15)	0.98(0.02)	0.97(0.02)	0.77(0.07)	0.89(0.03)	0.03(0.08)	0.04(0.12)	1.00(0.01)	1.00(0.01)
0.7	0.3	0.7	0.89(0.14)	0.91(0.10)	0.51(0.21)	0.55(0.19)	0.87(0.07)	0.85(0.06)	0.71(0.13)	0.84(0.06)	0.00(0.00)	0.00(0.00)	0.99(0.01)	0.99(0.01)
0.7	0.6	0.9	0.50(0.10)	0.59(0.08)	0.05(0.09)	0.05(0.10)	0.90(0.07)	0.89(0.05)	0.06(0.02)	0.35(0.06)	0.00(0.00)	0.00(0.00)	0.99(0.01)	0.99(0.01)
0.7	0.6	0.7	0.20(0.08)	0.29(0.08)	0.03(0.04)	0.04(0.04)	0.60(0.09)	0.59(0.08)	0.02(0.01)	0.25(0.02)	0.00(0.00)	0.00(0.00)	0.91(0.04)	0.99(0.04)

One can see that at small sizes with regarding ARI CESNA wins three times (out of 8); while if one considers NMI, CESNA wins two more settings. At all the other cases, including at medium size datasets, SEFNAC wins. SIAN never wins in this table. There is an impressive change in the performance of SIAN at the medium-sized datasets: SIAN comprehensively fails on all counts at medium sizes by producing NaN which we interpret as a one-cluster solution.

We also experimented with a slightly different design for categorical feature generation. That different design sets an entity to either coincide with its cluster center or to be entirely random. At that design CESNA wins 7 times at the small size datasets and SEFNAC wins at 7 medium size datasets.

Real world datasets lead to somewhat different results: CESNA performs rather poorly; SEFNAC wins three times regarding ARI and two times regarding NMI, and SIAN, two times regarding ARI and three times regarding NMI (see Table 4).

Here, we chose that data normalization method leading, on average, to the larger ARI values. Specifically, we used z-scoring for normalizing features in Lawyers dataset, HVR data set and COSN data set. The best results on World-Trade data set and parliament data set are obtained with no normalization. The network data in Lawyers and HVR are normalized with applying the modularity transformation [15]. The network data of COSN is normalized by shifting all the similarities to the average link value [13].

Table 4. Comparison of CESNA, SIAN and SEFNAC on Real-world data sets; average values of ARI and NMI and their standard deviation (std) are presented over 10 random initialisations. The best results are shown in bold-face.

Data sets / Alg.	CESNA		SIAN		SEFNAC	
	ARI	NMI	ARI	NMI	ARI	NMI
HRV6	0.20(0.00)	0.37(0.00)	0.39(0.29)	0.39(0.22)	0.45(0.14)	0.62(0.05)
Lawyers	0.28(0.00)	0.48(0.00)	0.59(0.04)	0.71(0.04)	0.63(0.06)	0.65(0.05)
World Trade	0.23(0.00)	0.59(0.00)	0.55(0.07)	0.77(0.03)	0.23(0.03)	0.58(0.04)
Parliament	0.25(0.00)	0.52(0.00)	0.79(0.12)	0.82(0.07)	0.28(0.01)	0.47(0.01)
COSN	0.44(0.00)	0.45(0.00)	0.43(0.05)	0.61(0.03)	0.50(0.11)	0.64(0.06)

5 Conclusion

This paper proposes a novel combined data recovery criterion for the problem of detecting communities in a feature-rich network. Our algorithm SEFNAC (Sequential Extraction of Feature-Rich Network Addition Clusters) extracts clusters one by one. Our approach is more or less universal regarding the scales of the data available. On the other hand, SEFNAC results may depend on data normalization.

We experimentally show that SEFNAC is competitive over both synthetic and real-world data sets against two popular state-of-the-art algorithms, CESNA [21] and SIAN [16].

Possible directions for future work:

- A systematic investigation of the relative effect of different data standardization methods on the results of SEFNAC.
- An extension of SEFNAC to large datasets should be proposed and validated.
- A trade-off between two constituent data sources, network and features, as expressed by factors λ and ξ , should be investigated.

References

1. Bojchevski, A., Günnemann, S.: Bayesian robust attributed graph clustering: joint learning of partial anomalies and group structure. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
2. Chiang, M.M.T., Mirkin, B.: Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads. *J. Classif.* **27**(1), 3–40 (2010)
3. Chunaev, P.: Community detection in node-attributed social networks: a survey (2019). arXiv preprint [arXiv:1912.09816](https://arxiv.org/abs/1912.09816)
4. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley, New York (2012)
5. Cross, R.L., Parker, A.: The Hidden Power of Social Networks: Understanding How Work Really Gets Done in Organizations. Harvard Business Press, Boston (2004)
6. Hubert, L., Arabie, P.: Comparing partitions. *J. Classif.* **2**(1), 193–218 (1985)
7. He, D., Jin, D., Chen, Z., Zhang, W.: Identification of hybrid node and link communities in complex networks. *Nat. Sci. Rep.* **5**, 8638 (2015)

8. Interdonato, R., Atzmueller, M., Gaito, S., Kanawati, R., Largeron, C., Sala, A.: Feature-rich networks: going beyond complex network topologies. *Appl. Netw. Sci.* **4**, 4:1–4:13 (2019)
9. Larremore, D.B., Clauset, A., Buckee, C.O.: A network approach to analyzing highly recombinant malaria parasite genes. *PLoS Comput. Biol.* **9**(10), e1003268 (2013)
10. Lazega, E.: *The Collegial Phenomenon: The Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership*. Oxford University Press, Oxford (2001)
11. Leskovec, J., Sosič, R.: SNAP: a general-purpose network analysis and graph-mining library. *ACM Trans. Intell. Syst. Technol. (TIST)* **8-1**, 1 (2016). CESNA on Github: <https://github.com/snap-stanford/snap/tree/master/examples/cesna>
12. Mirkin, B., Nascimento, S.: Additive spectral method for fuzzy cluster analysis of similarity data including community structure and affinity matrices. *Inf. Sci.* **183**(1), 16–34 (2012)
13. Mirkin, B.: *Clustering: A Data Recovery Approach*, 1st edn. (2005); 2d edn. (2012). CRC Press, Routledge (2005; 2012)
14. Nature Communications. <https://www.nature.com/articles/ncomms11863>
15. Newman, M.E.: Modularity and community structure in networks. *Proc. Nat. Acad. Sci.* **103**(23), 8577–8582 (2006)
16. Newman, M.E., Clauset, A.: Structure and inference in annotated networks. *Nat. Commun.* **7**, 11863 (2016)
17. De Nooy, W., Mrvar, A., Batagelj, V.: *Exploratory Social Network Analysis with Pajek*, chap. 2. Cambridge University Press, Cambridge (2004)
18. Stanley, N., Bonacci, T., Kwitt, R., Niethammer, M., Mucha, P.J.: Stochastic block models with multiple continuous attributes. *Appl. Netw. Sci.* **4**(1), 1–22 (2019)
19. Snijders, T.: The Siena webpage. https://www.stats.ox.ac.uk/~snijders/siena/Lazega_lawyers_data.htm
20. Xu, Z., Ke, Y., Wang, Y., Cheng, H., Cheng, J.: A model-based approach to attributed graph clustering. In: *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pp. 505–516. ACM (2012)
21. Yang, J., McAuley, J., Leskovec, J.: Community detection in networks with node attributes. In: *2013 IEEE 13th International Conference on Data Mining*, pp. 1151–1156. IEEE <https://arxiv.org/pdf/1401.7267.pdf> (2013). Accessed 22 Nov 2019



Efficient Community Detection by Exploiting Structural Properties of Real-World User-Item Graphs

Larry Yueli Zhang^(✉) and Peter Marbach

University of Toronto, Toronto, ON, Canada
{ylzhang,marbach}@cs.toronto.edu

Abstract. In this paper, we study the problem of detecting communities in real-world user-item graphs, i.e., bipartite graphs that represent interactions between a user and an item. Instead of developing a generic clustering algorithm for arbitrary graphs, we tailor our algorithm for user-item graphs by taking advantage of the inherent structural properties that exist in real-world networks. Assuming the existence of the core-periphery structure that has been experimentally and theoretically studied, our algorithm is able to extract the vast majority of the communities existing in the network by performing dramatically less computational work compared to conventional graph-clustering algorithms. The proposed algorithm achieves a subquadratic runtime (with respect to the number of vertices) for processing the entire graph, which makes it highly practical for processing large-scale graphs which typically arise in real-world applications. The performance of the proposed algorithm, in terms of both community-detection accuracy and efficiency, is experimentally evaluated with real-world datasets.

Keywords: Community detection · User-item graphs · Structural properties · Social networks

1 Introduction

Numerous real-world Internet applications generate large amounts of data consisting of “user-item” interactions. For example, in video streaming services such as YouTube and Netflix, users interact with the videos by watching or rating them; in an online shopping application such as Amazon, users interact with the items by viewing or purchasing them; in a social network application such as Twitter, the follower/following relation can also be modelled as user-item interactions. It is an important problem to detect the underlying community structure of these user-item interactions. More precisely, we want to detect a set of users that share a common interest and, as a result, tend to interact with a common set of items. Being able to identify such communities is essential for functionalities such as item recommendation and discovering similar-minded users. The data of user-item interactions can be modelled as a bipartite user-item graph

and the task of detecting a community involves identifying a user set along with an item set, where the user set consists of users who share a common interest, and the item set consists of items that are representative of the shared interest.

The community detection problem has traditionally been modelled as a problem of finding clusters of densely connected vertices (e.g., cliques and quasi-cliques) in a graph. There has been a rich literature on graph clustering algorithms. Compared to most of the existing works, a key difference in the design of the proposed algorithm is that it attempts to exploit the special structural properties that are inherent in real-world social graphs. That is, rather than being a generic clustering algorithm for arbitrary graphs, the proposed algorithm is designed to be efficient for a particular subset of graphs – user-item graphs that are formed by human agents interacting with items of their interest. We focus our effort on detecting “interest-based communities”, i.e., a group of individuals that share a common interest. The structural property that the algorithm takes into consideration is the “core-periphery” structure, i.e., the existence of “cores users” whose interests are highly identical to the common interest of the community. By specializing to detect such core users, the proposed algorithm is able to effectively detect all communities in the graph by examining a much smaller search space than that of an exhaustive generic clustering algorithm, therefore achieves a dramatically high efficiency in terms of the time needed to process the entire graph. More precisely, the proposed algorithm achieves a time complexity that is guaranteed to be below quadratic time (in terms of the number of vertices) and is practically close to linear-time with certain datasets. Our experiments (presented in Sect. 6) show that the actual runtime of the proposed algorithm on large graphs is several magnitudes shorter compared to existing community detection algorithms for bipartite graphs. It also effectively detects the vast majority of communities that can be detected by other algorithms.

In summary, the contributions of this paper are to (1) propose the design of a low-complexity community detection algorithm that is tailored for real-world user-item graphs by exploiting assumptions on the structural properties of the graph, (2) provide experimental results on real-world datasets to investigate the efficiency and accuracy of the algorithm, and (3) demonstrate the effectiveness of the algorithm-design method that specializes the algorithm to take advantage of structures in real-world complex networks.

2 Related Work

Community structure and community detection algorithms in networked systems has long been an important topic in many fields of research such as biology, physics, economics, and computer science. A comprehensive survey of the early works is provided in [10], which we cite in lieu of having an exceedingly long list of references. Community detection algorithms that are designed for bipartite graphs [10] fall into two categories: spectrum-based (such as methods based on matrix factorization [5]) and modularity-based [2,3]. These algorithms require global information of the entire input graph to compute the

spectrum or modularity, therefore do not scale well when the size of the input graph becomes substantially large. Algorithms based on the detection of quasi-cliques/bicliques [1, 7, 12, 14, 16, 19] are the state of the art. The best known runtime of quasi-clique algorithms is $\mathcal{O}(|V|^3)$ (e.g., in [12]) for processing the entire graph, where $|V|$ is the number of vertices in the graph.

The algorithm design in this paper is based on efficiently taking advantage of the core-periphery structure [6, 8, 13, 15, 17, 20, 21] in real-world social graphs, i.e., the existence of densely connected subgraphs surrounded by sparsely connected periphery nodes. Communities in social graphs exhibit internal core-periphery structure. In [17] it is explained that the core component can be viewed as users who are highly identical in their interests and the peripheries are the users whose interests partially overlap with the core. A key idea behind the proposed algorithm in this paper is to detect communities by identifying their core users.

3 Intuition Behind the Algorithm

In this section, we provide the main intuition behind the algorithm that we propose, i.e., we discuss the structural properties of communities in the user-item graphs that we use to define our algorithm. For this, we focus on the so-called “interest-based” community which is given by a group of individuals that share a common interest. We refer to the common interest of the community as the core interest of the community. Note that the communities in a user-item graph are interest-based communities, i.e., a community consists of a set of users that have the same interest in the sense they are interested in the same type of items.

Interest-based communities have been extensively studied in the literature [11, 13, 18, 21]. An important result is the existence of the core-periphery structure [6, 8, 13, 15, 17, 20, 21]. That is, for a given community, there exists a set of core users whose interests are largely identical to the core interest of the community, and there exist peripheral users who have some overlap with the core interest but the overlap is partial. This result is important as it implies that we can identify a community as a tuple consisting of (a) the core interest and (b) the core users of the community. Moreover, given the core interest of a community, we can identify the core users by selecting the set of users that, not only share the core interest, but also have the core interest as their main interest. Similarly, given a set of core users, we can identify the core interest of the community by finding the set of topics that all (or a large portion of) core users are interested in while users outside the core are only partially interested in.

In a user-item graph, an interest-based community can be identified by a tuple consisting of (a) the core items and (b) the core users of the community. The core items of the community are the items that represent the core interest of the community. This relationship between the core items and the core users is the key property that we use to design our algorithm. To highlight this, we use the terms “popularity” and “typicality” to describe this property. More precise definitions of these terms can be found in Sect. 4.

4 Model

A *user-item graph* is an undirected bipartite graph in which the vertices are divided into two sets, the *user set* and the *item set*. Let U denote the set of all user vertices and I be the set of all item vertices. An edge (u, i) , $u \in U, i \in I$ exists if user u interacts with/accesses item i . The bipartite user-item graph is a natural model for a wide range of real-life applications. For example, in video streaming services such as YouTube and Netflix, the user-video viewing/rating relationship can be modelled by a user-item graph.

Based on the discussion in Sect. 3, an interest-based community is defined by a set of core users who share a common interest identified by a set of core items. More precisely, given the core users of a community, we say that the core items of the community are the items that are (a) *popular* among the core users in the sense that a core item is accessed/connected by a significant fraction of the core users, and (b) *typical* among the core users in the sense that a significant fraction of the accesses/edges of a core item come from within the core users instead of from outside the core users. Similarly, given the core items of a community, we say that the core users of the community are the users that are (a) *popular* among the core items in the sense that a core user accesses a significant fraction of the core items, and (b) *typical* among the core items in the sense that a significant fraction of the accesses made by a core user go to *within* the core items instead of *outside* the core items. Below are the formal definitions.

Definition 1 Given an item $i \in I$ and a user set $U_C \subseteq U$. Let U_i be the set of users that interact with item i . The **popularity** of item i over U_C is $\rho(i, U_C) = |U_i \cap U_C| / |U_C|$ and the **typicality** of item i over U_C is $\epsilon(i, U_C) = |U_i \cap U_C| / |U_i|$

Definition 2 Given a user $u \in U$ and an item set $I_C \subseteq I$. Let I_u be the set of items that user u interacts with, the **popularity** of user u over I_C is $\alpha(u, I_C) = |I_u \cap I_C| / |I_C|$ and the **typicality** of user u over I_C is $\delta(u, I_C) = |I_u \cap I_C| / |I_u|$

For the proposed algorithm, the *community* is represented by the combination of a core user set and a core item set. We require that each user and item in the core has lower-bounded popularity and typicality. Below is a formal definition.

Definition 3 A $(\bar{\delta}, \bar{\epsilon}, \bar{\rho}, \bar{\alpha})$ -**community** consists a user set $U_C \subseteq U$ and an item set $I_C \subseteq I$ of items such that $\forall u \in U_C, \delta(u, I_C) \geq \bar{\delta} \wedge \alpha(u, I_C) \geq \bar{\alpha}$ and $\forall i \in I_C, \epsilon(i, U_C) \geq \bar{\epsilon} \wedge \rho(i, U_C) \geq \bar{\rho}$.

5 Algorithm

Based on the above definitions, we devise a community detection algorithm that detects $(\bar{\delta}, \bar{\epsilon}, \bar{\alpha}, \bar{\rho})$ -communities with given thresholds.

Algorithm 1: DETECT-SINGLE-COMMUNITY

Input: U_C : an initial set of users, $\bar{\delta}, \bar{\epsilon}, \bar{\alpha}, \bar{\rho}$
Output: (U_C, I_C) : a pair of user and item sets that is the core of a detected community, or NIL if a community is not detected

```

1 converged  $\leftarrow$  False
2 num_iterations  $\leftarrow$  0
3 while not converged and num_iterations < max_iterations do
4    $U'_C \leftarrow$  copy of  $U_C$ 
5    $I_C \leftarrow$  SELECT-ITEMS( $U_C, \bar{\epsilon}, \bar{\rho}$ )
6    $U_C \leftarrow$  SELECT-USERS( $I_C, \bar{\delta}, \bar{\alpha}$ )
7   if  $U_C = U'_C$  then
8      $\left|$  converged  $\leftarrow$  True
9      $\left|$  num_iterations  $\leftarrow$  num_iterations + 1
10 if converged then
11    $\left|$  return  $(U_C, I_C)$ 
12 else
13    $\left|$  return NIL

```

Algorithm 1 shows the routine for detecting a single community core. Given the user set, the SELECT-ITEMS subroutine returns the set of items that satisfy the constraints on the item's popularity and typicality with respect to the user set. The SELECT-USERS works symmetrically in a similar way.

Algorithm 2: DETECT-ALL-COMMUNITIES

Input: the user-item graph with user set U and item set I , $\bar{\delta}, \bar{\epsilon}, \bar{\alpha}, \bar{\rho}$
Output: C : a set of communities each being a pair of user and item sets

```

1  $C \leftarrow \emptyset$ 
2 foreach item  $i$  in  $I$  do
3    $U_i \leftarrow$  the set of users that are connected with  $i$ 
4    $c \leftarrow$  DETECT-SINGLE-COMMUNITY( $U_i, \bar{\delta}, \bar{\epsilon}, \bar{\alpha}, \bar{\rho}$ )
5   if  $c$  is not NIL then
6      $\left|$  Add  $c$  to  $C$ 
7 return  $C$ 

```

The runtime of DETECT-SINGLE-COMMUNITY depends on the exact topology of the input graph. For simplicity, while estimating the correct order of magnitude, we perform our runtime analysis by assuming an average-case graph topology where the degrees of the vertices, as well as the sizes of the communities, are

uniform. In the average-case graph, let d_I be the average number of (user) neighbours of an item and d_U be the average number of (item) neighbours of a user. The number of users and items being selected in each iteration is upper-bounded by a constant in our algorithm’s implementation. The number of the candidate items that need to be processed in **SELECT-ITEMS** is in $\mathcal{O}(d_U)$. The set intersection performed when computing the popularity/typicality is between a set of size d_U and a set of constant size. Therefore, the runtime for **SELECT-ITEMS** is overall $\mathcal{O}(d_U)$. Similarly, the overall runtime of **SELECT-USERS** is $\mathcal{O}(d_I)$. The main loop in Algorithm 1 typically terminates after a small constant number of iterations (as will be shown experimentally in Sect. 6). Therefore, the overall runtime of **DETECT-SINGLE-COMMUNITY** is in the order of $\mathcal{O}(d_U + d_I)$. More precisely, let $|V| = |U| + |I|$ be the total number of vertices in the graph and d_U and d_I be functions of $|V|$, the above runtime is now $\mathcal{O}(d_U(|V|) + d_I(|V|))$.

Algorithm 1 detects a single community core from an initial user set. To detect all communities in a given user-item graph, theoretically, it would suffice if we run **DETECT-SINGLE-COMMUNITY** on all possible subsets of the user set U of the graph. However, it would lead to exponential runtime. The proposed algorithm does the following: for each item, use its neighbouring user set as an initial set to detect a community. In this way, we only need to run Algorithm 1 on $|I|$ initial user sets. Based on the core-periphery structure of real-world interest-based communities, it can be argued that only going through these initial sets would be sufficient for detecting all the communities/interests in the graph: assuming that each community contains a set of core users/items who are dedicated to the interest of the community, it suffices to detect all the community cores in order to detect all the communities. Moreover, because the core users/items are highly focussed on the interest of the community, an iteration starting from the user set of a core item is highly likely to converge to the core itself. This observation is critical for reducing the overall runtime of processing the entire user-item graph. Algorithm 2 is the pseudocode of the routine for detecting all communities.

The time complexity of Algorithm 2 is simply $|I|$ multiplied by the runtime of Algorithm 1, i.e., $\mathcal{O}(|I| \cdot (d_U(|V|) + d_I(|V|)))$. Let $d(|V|) = \max(d_U(|V|), d_I(|V|))$, and given that $|V| = |U| + |I|$, the overall runtime of Algorithm 2 becomes $\mathcal{O}(|V| \cdot d(|V|))$. Since the degree of a vertex is upper-bounded by $|V|$, the overall runtime for detecting all communities is guaranteed to be in $\mathcal{O}(|V|^2)$.

6 Experimental Evaluation

We perform our experiments using two real-world datasets: the Netflix dataset and the Yelp dataset. The Netflix dataset [4] consists of 100,480,507 ratings that 480,189 users gave to 17,770 movies, which can be modelled as a user-item graph with each rating as a user-item edge (ignoring the value of the rating). The Yelp Dataset [9] contains user reviews of businesses posted on the Yelp across 10 metropolitan areas. We select the subgraph of one metropolitan area (Toronto, Ontario). This results in a dataset with 148,570 users, 33,412 businesses, and 784,462 reviews. Compared to the Netflix dataset, the Yelp dataset is “sparser” in the sense that the average degree of a vertex is much lower.

6.1 Evaluation on Detected Communities

We compare the proposed algorithm (namely MUISI, standing for Mutual User-Item Subset Iterations) with a state-of-the-art quasi-biclique algorithm in [12] (named LIU hereinafter). The definition of a quasi-biclique in LIU is closely related to MUISI—it is essentially the MUISI definition with only the popularity constraints. In order for LIU to finish in a reasonable amount of time, we sampled a subset of the Netflix data in the following way: pick 200 movies from a number of known communities, then randomly add 100 additional movies as “noise” items. Among all users that rated any of the 300 movies, we selected uniformly at random a subset of 3000 users. The resulting graph contains 14,540 edges.

Table 1. Example communities detected by MUISI and LIU.

ID	Year	Title	# of users
5907	1956	Godzilla: King of the Monsters	1033
15810	1964	Godzilla vs. Mothra	916
409	1966	Godzilla vs. The Sea Monster	447
14623	1971	Godzilla vs. Hedorah	327
12506	1974	Godzilla vs. Mechagodzilla	371
17746	1991	Godzilla & Mothra: Battle for Earth	974
8656	1993	Godzilla vs. Mechagodzilla II	470
10642	1999	Godzilla 2000: Millennium	1926
8824	2001	Godzilla, Mothra and King Ghidorah	1088
4461	2002	Godzilla Against Mechagodzilla	791

(a) The “Godzilla”-related community detected by MUISI.

ID	Year	Title	# of users
872	1954	Seven Samurai	31691
15431	1954	Creature from the Black Lagoon	3632
4489	1961	Mysterious Island	1404
5538	1979	Monty Python’s Life of Brian	43630
17746	1991	Godzilla & Mothra: Battle for Earth	974
6001	1992	Tom and Jerry: The Movie	1698
11283	1994	Forrest Gump	181508
1173	1999	Walking with Dinosaurs	3867
8824	2001	Godzilla, Mothra and King Ghidorah	1088

(b) The “Godzilla”-related community detected by LIU.

Table 1 shows a comparison between the communities produced by MUISI and LIU (with a popularity threshold of 50% for both algorithms and a typicality threshold of 10% for MUISI). Due to the length limit, we only present one typical

pair of communities as an example. The two algorithms produce two different “flavours” of communities. The community detected by LIU tends to include movies with a larger number of users, which is expected because it only has the popularity constraint. The movies in the MUISI community have a relatively smaller number of users and, because of the typicality constraint, are more closely related to the specific interest. In contrast, the items in the LIU community are more loosely related and can introduce “noisy” items (such as “Forest Gump”) which are highly popular but are not typical for the community.

We are interested in whether the two algorithms detect “equivalent” sets of communities. To make it concrete, we consider the application scenario where we use the communities to provide content recommendation. We say the two sets of communities are equivalent if they function equivalently in content recommendation. For each user, we look at its set of visited items and “decompose” this item set by taking the intersection of it with the core item set of every detected community. As a result, we obtain an “interest vector” for each user. Each component of the interest vector $\{v_c\}$ is defined as follows. Let I_u be the item set of user u and let I_c be the recommended item set of a community c (computed using the core user set), and C is the set of all detected community cores, $v_c = I_u \cap I_c, \forall c \in C$. To verify that the MUISI communities cover the majority of the LIU communities, we compute for each user their interest vectors based on both the MUISI and LIU communities. For each non-empty component v_c of the LIU vector, we check if there exists a v'_c in the MUISI vector that is identical to v_c . If yes, we say v_c is covered. The only parameter in this comparison is the recommendation popularity that is used to obtain the recommended item sets I_c (this recommendation can be replaced with other mechanisms depending on the application). We check this coverage in both directions, i.e., we check how much of the LIU communities are covered by the MUISI communities as well as the percentage of MUISI communities covered by LIU.

Table 2. Interest vector coverage between LIU and MUISI (Netflix data)

Recommend Popularity	0.1	0.2	0.3	0.4	0.5
MUISI cover LIU (%)	99.8	97.6	92.2	87.8	81.5
LIU cover MUISI (%)	96.8	95.4	94.9	94.3	93.2

(a) Detection popularity threshold = 0.1

Recommend Popularity	0.1	0.2	0.3	0.4	0.5
MUISI cover LIU (%)	99.8	97.6	92.2	87.8	81.5
LIU cover MUISI (%)	96.8	95.4	94.9	94.3	93.2

(b) Detection popularity threshold = 0.5

Table 3. Interest vector coverage between LIU and MUISI (Yelp data)

Recommend Popularity	0.1	0.2	0.3	0.4	0.5
MUISI cover LIU (%)	98.5	98.4	98.4	98.1	90.1
LIU cover MUISI (%)	91.3	91.3	89.9	89.8	84.6

(a) Detection popularity threshold = 0.1

Recommend Popularity	0.1	0.2	0.3	0.4	0.5
MUISI cover LIU (%)	99.9	99.4	99.1	99.1	99.0
LIU cover MUISI (%)	99.1	98.9	98.8	98.2	95.9

(b) Detection popularity threshold = 0.5

Table 2 shows the covering percentages (averaged over all users) under different detection and recommendation popularity. The mutual coverage between

MUISI communities and LIU communities is above 80% in most cases. The coverage only dropped slightly below 80% when we apply a high recommendation popularity (0.5) with communities detected using low popularity (0.1). Table 3 presents the same comparison using the Yelp dataset. The coverage is overall higher than the Netflix result. This gives us the insight that the Yelp dataset contains “clearer” communities. We conclude that the MUISI and LIU communities achieve similar outcome in content recommendation; in other words, the MUISI communities are equivalent to those detected by LIU.

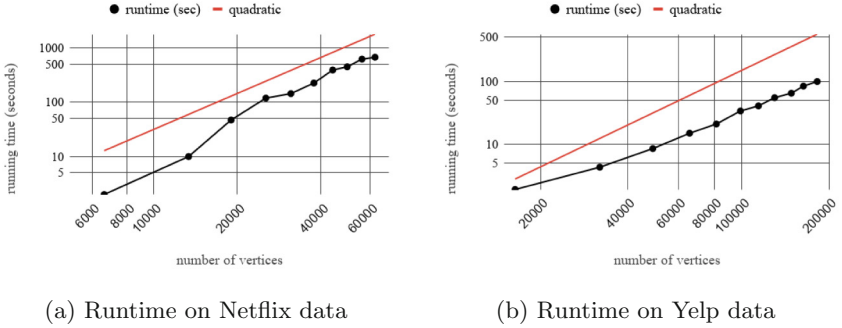


Fig. 1. Asymptotic runtime evaluation of MUISI

6.2 Evaluation on Runtime

We compare the time efficiency of the MUISI algorithm with the quasi-clique algorithm LIU [12]. We generated sampled subgraphs of the Netflix dataset and Yelp dataset of varying sizes as inputs and compared the time taken to process the entire input. All experiments were run on a computer with 2.7GHz Intel Core i5 CPU, 8 GB 1867MHz DDR3 RAM. As shown in Table 4, MUISI

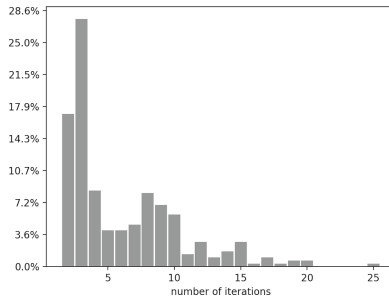


Fig. 2. Histogram of the number of subset iterations taken before convergence.

Table 4. Runtime comparison between MUISI and LIU

# items	20	40	60	80	100	# items	100	200	400	600	800
# users	237	570	720	838	1250	# users	434	993	1718	2615	3603
# edges	294	957	1487	2053	3104	# edges	479	1247	2197	3737	5471
MUISI (ms)	90	180	300	320	700	MUISI (ms)	640	710	730	780	860
LIU (ms)	360	6,470	13,290	28,980	183,140	LIU (ms)	1,360	9,480	22,330	41,930	336,760

(a) Comparison with the Netflix dataset

(b) Comparison with the Yelp dataset

has a significantly lower runtime and scales much better than the quasi-clique algorithm as the input size increases.

We also evaluated the asymptotic runtime of the proposed algorithm using both the Netflix and Yelp datasets sampled in varying sizes. For the sampling, we randomly choose a subgraph with desired numbers of items and users such that the item-vs-user ratio stays similar to that of the original graph. The result is plotted in Fig. 1 with log-log scaled axes. The slope of the red line is the reference quadratic growth. For both datasets, the growth of the runtime is either close to or slower than the quadratic growth rate. This asymptotic pattern agrees with our theoretical analysis on the algorithm’s complexity.

6.3 Evaluation on Convergence

We used the MUISI algorithm to process the complete Netflix dataset with a popularity threshold of 0.2 and a typicality threshold of 0.02, and a total of 4,617 communities were detected. We are interested in the number of iterations taken before each DETECT-SINGLE-COMMUNITY routine converges. Figure 2 is the histogram that shows the distribution this number. The distribution is bimodal in the sense that it has two peak areas around 4 and 8. Our hypothesis is that, when a seed item belongs to the core of some ground-truth community, the iterations would converge very quickly (within 4 iterations); whereas when the item does not belong to a community core, it would take a longer time (around 8 iterations). We verified this hypothesis by reviewing the output and dividing the iterations according to whether the seed item ends up in a community core. As a result, the average number of iterations from a core item is 3.2 while the average for non-core items is 6.8, which agrees with our hypothesis.

Another hypothesis to verify is that, when starting from a core item of a community, the iterations are highly likely to converge to the core it is from. This hypothesis would imply that, as long as a community core exists, it is guaranteed to be detected by the algorithm. We re-ran iterations from the core items and, for each community core, we record the resulting item set with the maximum overlap with the starting community core. The overlap between two sets A and B is calculated as $|A \cap B|/|A \cup B|$. Figure 3 shows the distribution of the maximum overlap. It is evident that the majority of communities cores have items that would lead to iterations converging to themselves.

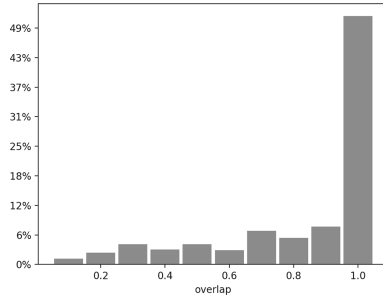


Fig. 3. Histogram of the maximum overlap between a community core’s item set and the resulting item set of iterations starting from one of its core items.

7 Conclusions

Algorithms that take advantage of the inherent structures of real-world complex networks can be surprisingly simple and efficient. Real-life social graphs, being generated by human agents that are motivated by specific objectives, naturally exhibit structural properties such as the core-periphery structure. The key idea behind the design of the proposed algorithm is to achieve performance by specializing the algorithm for a targeted category of inputs with such structures. The proposed algorithm achieves a significant improvement in the time complexity to process the entire input graph, reducing it to guaranteed subquadratic time. In addition to the popularity constraint that’s commonly used in existing methods, the proposed algorithm applies a second constraint on the typicality when detecting communities. In our experiments, the algorithm demonstrates a runtime that is several magnitudes shorter compared to the state-of-the-art quasi-clique based algorithms; in the meantime, it can detect communities that are equivalent to those detectable by other algorithms. These qualities make the proposed algorithm highly practical for real-world applications with large-scale graphs.

References

1. Abello, J., Resende, M.G., Sudarsky, S.: Massive quasi-clique detection. In: Latin American Symposium on Theoretical Informatics, pp. 598–612. Springer (2002)
2. Barber, M.J.: Modularity and community detection in bipartite networks. *Phys. Rev. E* **76**(6), 066102 (2007)
3. Beckett, S.J.: Improved community detection in weighted bipartite networks. *R. Soc. Open Sci.* **3**(1), 140536 (2016)
4. Bennett, J., Lanning, S., et al.: The netflix prize. In: Proceedings of KDD Cup and Workshop, New York, vol. 2007, p. 35 (2007)
5. Bokde, D., Girase, S., Mukhopadhyay, D.: Matrix factorization model in collaborative filtering algorithms: A survey. *Proc. Comput. Sci.* **49**, 136–146 (2015)
6. Borgatti, S.P., Everett, M.G.: Models of core/periphery structures. *Soc. Netw.* **21**(4), 375–395 (2000)

7. Brunato, M., Hoos, H.H., Battiti, R.: On effectively finding maximal quasi-cliques in graphs. In: International Conference on Learning and Intelligent Optimization, pp. 41–55. Springer (2007)
8. Csermely, P., London, A., Wu, L.Y., Uzzi, B.: Structure and dynamics of core/periphery networks. *J. Complex Netw.* **1**(2), 93–123 (2013)
9. Dataset, T.Y.: <https://www.yelp.ca/academic.dataset>. Accessed July 2018
10. Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**(3–5), 75–174 (2010)
11. Li, X., Guo, L., Zhao, Y.E.: Tag-based social interest discovery. In: Proceedings of the 17th International Conference on World Wide Web, pp. 675–684. ACM (2008)
12. Liu, X., Li, J., Wang, L.: Modeling protein interacting groups by quasi-bicliques: complexity, algorithm, and application. *IEEE/ACM Trans. Comput. Biol. Bioinf. (TCBB)* **7**(2), 354–364 (2010)
13. Marbach, P.: The structure of communities in information networks. In: Information Theory and Applications Workshop (ITA), 2016, pp. 1–6. IEEE (2016)
14. Pattillo, J., Veremyev, A., Butenko, S., Boginski, V.: On the maximum quasi-clique problem. *Discrete Appl. Math.* **161**(1–2), 244–257 (2013)
15. Rombach, M.P., Porter, M.A., Fowler, J.H., Mucha, P.J.: Core-periphery structure in networks. *SIAM J. Appl. Math.* **74**(1), 167–190 (2014)
16. Sim, K., Li, J., Gopalkrishnan, V., Liu, G.: Mining maximal quasi-bicliques: novel algorithm and applications in the stock market and protein networks. *Statistical Anal. Data Min. ASA Data Sci. J.* **2**(4), 255–273 (2009)
17. Yang, J., Leskovec, J.: Overlapping communities explain core-periphery organization of networks. *Proc. IEEE* **102**(12), 1892–1902 (2014)
18. Yang, J., Leskovec, J.: Defining and evaluating network communities based on ground-truth. *Knowl. Inf. Syst.* **42**(1), 181–213 (2015)
19. Zeng, Z., Wang, J., Zhou, L., Karypis, G.: Coherent closed quasi-clique discovery from large dense graph databases. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 797–802. ACM (2006)
20. Zhang, J., Ackerman, M.S., Adamic, L.: Expertise networks in online communities: structure and algorithms. In: Proceedings of the 16th International Conference on World Wide Web, pp. 221–230. ACM (2007)
21. Zhang, L.Y., Marbach, P.: Stable and efficient structures for the content production and consumption in information communities. In: Game Theory for Networking Applications, pp. 163–173. Springer (2019)



Measuring Proximity in Attributed Networks for Community Detection

Rinat Aynulin^{1,3(✉)} and Pavel Chebotarev²

¹ Moscow Institute of Physics and Technology, 9 Institutskii per.,
141700 Dolgoprudny, Moscow Region, Russia
rinat.aynulin@phystech.edu

² Trapeznikov Institute of Control Sciences of the Russian Academy of Sciences,
65 Profsoyuznaya str., 117997 Moscow, Russia
pavel4e@gmail.com

³ Kotelnikov Institute of Radio-Engineering and Electronics (IRE) of the Russian
Academy of Sciences, Mokhovaya 11-7, 125009 Moscow, Russia

Abstract. Proximity measures on graphs have a variety of applications in network analysis, including community detection. Previously they have been mainly studied in the context of networks without attributes. If node attributes are taken into account, however, this can provide more insight into the network structure. In this paper, we extend the definition of some well-studied proximity measures to attributed networks. To account for attributes, several attribute similarity measures are used. Finally, the obtained proximity measures are applied to detect the community structure in some real-world networks using the spectral clustering algorithm.

Keywords: Attributed networks · Community detection · Proximity measure · Kernel on graph

1 Introduction

Many real-world systems from the fields of social science, economy, biology, chemistry, etc. can be represented as networks or graphs¹ [7]. A network consists of nodes representing objects, connected by edges representing relations between the objects. Nodes can often be divided into groups called clusters or communities. Members of such a cluster are more densely connected to each other than to the nodes outside the cluster.

The task of finding such groups is called clustering or community detection. There have been plenty of algorithms proposed by researchers in the past to address this problem.

Some of the community detection algorithms require the introduction of distance or proximity measure on the set of graph nodes: a function, which shows, respectively, the distance or proximity (similarity) between a pair of nodes. Only

¹ Formally, graph is a mathematical representation of a network. However, hereinafter, the terms “graph” and “network” will be used interchangeably.

the shortest path distance had been studied for a long time. Nowadays, we have a surprising variety of measures on the set of graph nodes [9, Chap.15]. Some of the proximity measures can be defined as kernels on graphs, i.e., symmetric positive semidefinite matrices [1].

Previously, kernels have been applied mainly to analyze networks without attributes. However, in many networks, nodes are associated with attributes that describe them in some way. Thereby, multiple dimensions of information can be available: a structural dimension representing relations between objects, a compositional dimension describing attributes of particular objects, and an affiliation dimension representing the community structure [3]. Combining information about relations between nodes and their attributes provides a deeper understanding of the network structure.

Many methods for community detection in attributed networks have been proposed recently. Surveys [3,6] describe existing approaches to this problem. We provide some information on this in Sect. 2. However, kernel-based clustering, as already noted, has not yet been applied to attributed networks.

In this paper, we extend the definition of a number of previously defined proximity measures to the case of networks with node attributes. Several similarity measures on attributes are used for this purpose. Then, we apply the obtained proximity measures to the problem of community detection in several real-world datasets.

According to the results of our experiments, taking both node attributes and node relations into account can improve the efficiency of clustering in comparison with clustering based on attributes only or on structural data only. Also, the most effective attribute similarity measures in our experiments are the Cosine Similarity and Extended Jaccard Similarity.

2 Related Work

This section is divided into two parts. In the first one, we provide a quick overview of papers where various measures on the set of graph nodes are discussed. Then, we introduce a few studies focused on community detection in attributed networks.

For a long time, only the shortest path distance has been widely used [10]. [9, Chap.15] provides a survey of dozens of measures that have been proposed in various studies in the last decades. Among them there are inspired by physics Resistance (also known as Electric) measure [31], logarithmic Walk measure discussed in [5], the Forest measure related to Resistance [4], and many others.

In [1], the authors analytically study properties of various proximity measures² and kernels on graphs, including Walk, Communicability, Heat, PageRank, and several logarithmic measures. Then, these measures are compared in the context of spectral clustering on the stochastic block model. [12] provides a survey and numerical comparison of nine kernels on graphs in application to link prediction and clustering problems.

² Here, we use the term “proximity measure” in a broadened sense and, unlike [1], do not require a proximity measure to satisfy the triangle inequality for proximities.

In [33], the authors numerically study the efficiency of the Corrected Commute-Time, Free Energy, Logarithmic Forest, Randomized Shortest-Path, Sigmoid Commute-Time, and Shortest-Path measures in experiments with 15 real-world datasets. In [2, 16] it was proposed to improve the efficiency of some existing proximity measures by applying simple mathematical functions like logarithm to them.

Classically, community detection algorithms used either structure information (see, e.g., [14]) or information about node attributes (e.g., [17]). Recently, the idea of detecting communities based both on the structure and attribute data has attracted a lot of attention. Taking into account that it is possible to consider also edge attributes, we will focus on the attributes of nodes.

In [37], the authors proposed the SA-Clustering algorithm. The idea of the algorithm is the following: first, an attribute node is created for each value of each attribute. An attribute edge is drawn between the “real” node and attribute node if the node has the value of the attribute specified in the attribute node. The random walk model then is used to estimate the distance between nodes. Communities are determined using the k -medoids method. The CODICIL method is presented in [29]. This method adds content edges as a supplement to structure edges. The presence of a content edge between two nodes means the similarity of the node attributes. Then, the graph with content edges is clustered using the Metis and Multi-level Regularized Markov Clustering algorithms.

Reference [26] proposes the method for community detection in attributed networks based on weight modification. For every existing edge, the weight of the edge is assigned to the matching coefficient between the nodes. This coefficient equals to the number of attribute values the nodes have in common. The network with modified edges is clustered using the Karger’s Min-Cut, MajorClust, and Spectral algorithms. In [36], the CESNA method is proposed. This method assumes the attributed networks to be generated by a probabilistic model. Communities are detected using maximum-likelihood estimation on this model.

For a more detailed review of recently proposed methods for community detection in attributed networks, see [3, 6]. Within the classification presented in [6], our approach belongs to the class of weight-based methods.

3 Background and Preliminaries

3.1 Definitions

Let $G = (V, E, F)$ be an undirected weighted attributed graph with the set of nodes V ($|V| = n$), the set of edges E ($|E| = m$), and the tuple of attribute (or feature) vectors F . Each of the n nodes is associated with d attributes, so $F = (\mathbf{f}_1, \dots, \mathbf{f}_n)$, where $\mathbf{f}_i \in \mathbb{R}^d$. In the experiments, we will consider networks with binary attributes.

The *adjacency matrix* A of the graph is a square matrix with elements a_{ij} equal to the weight of edge (i, j) if node i is connected to node j and equal to zero otherwise. In some applications, each edge can also be associated with a positive value

c_{ij} , which is the cost of following this edge. If cost does not appear naturally, it can be defined as $c_{ij} = \frac{1}{a_{ij}}$. The *cost matrix* C contains costs of all the edges.

The *degree* of a node is the sum of the weights of the edges linked to the node. The diagonal *degree matrix* $D = \text{diag}(A \cdot \mathbf{1})$ shows degrees of all the nodes in the graph ($\mathbf{1} = (1, \dots, 1)^T$). Given A and D , the *Laplacian matrix* is defined as $L = D - A$, and the *Markov matrix* is $P = D^{-1}A$.

A *measure* on the set of graph nodes is a function κ that characterizes proximity or similarity between the pairs of graph nodes. A *kernel on graph* is a similarity measure that has a Gram matrix (symmetric positive semidefinite matrix) representation K . Given K , the corresponding distance matrix Δ can be obtained from the equation

$$K = -\frac{1}{2}H\Delta H, \quad (1)$$

where $H = I - \frac{1}{n}\mathbf{1} \cdot \mathbf{1}^T$.

For more details about graph measures and kernels, we refer to [1].

3.2 Community Detection Algorithms

***k*-means.** The *k*-means algorithm [24] is used in this study for community detection based on the attribute information.

Spectral. In this paper, we use the variation of the Spectral algorithm presented by Shi and Malik in [32]. The approach is based on applying the *k*-means algorithm to the eigenvectors of the Laplacian matrix of the graph. For a detailed review of the mathematics behind the Spectral algorithm, we refer to the tutorial by Ulrike von Luxburg [22].

3.3 Measures

In this study, we consider five measures which have shown a good efficiency in [1, 33].

Communicability. $K^C = \sum_{n=0}^{\infty} \frac{\alpha^n A^n}{n!} = \exp(\alpha A)$, $\alpha > 0$ [11, 13].

Heat. $K^H = \sum_{n=0}^{\infty} \frac{\alpha^n (-L)^n}{n!} = \exp(-\alpha L)$, $\alpha > 0$ [19].

PageRank. $K^{\text{PR}} = (I - \alpha P)^{-1}$, $0 < \alpha < 1$ [12, 27].

Free Energy. Given P , C and the parameter α , the matrix W can be defined as $W = \exp(-\alpha C) \circ P$ (the “ \circ ” symbol stands for element-wise multiplication). Then, $Z = (I - W)^{-1}$ and $S = (Z(C \circ W)) \div Z$ (the “ \div ” symbol stands for element-wise division). Finally, $\Delta^{\text{FE}} = \frac{\Phi + \Phi^T}{2}$, where $\Phi = \frac{\log(Z)}{\alpha}$. K^{FE} can be obtained from Δ^{FE} using transformation (1) [18].

Sigmoid Corrected Commute-Time. First, let us define the Corrected Commute-Time (CCT) kernel: $K^{\text{CCT}} = HD^{-\frac{1}{2}}M(I - M)^{-1}MD^{-\frac{1}{2}}H$, where $H = I - \frac{\mathbf{1}\mathbf{1}^T}{n}$, $M = D^{-\frac{1}{2}}(A - \frac{\mathbf{d}\mathbf{d}^T}{\text{vol}(G)})D^{-\frac{1}{2}}$, \mathbf{d} is a vector of elements of the diagonal degree matrix D , $\text{vol}(G) = \sum_{ij=1}^n a_{ij}$. Then, the elements of K^{SCCT} are equal to $K_{ij}^{\text{SCCT}} = \frac{1}{1 + \exp(-\alpha K_{ij}^{\text{CCT}}/\sigma)}$, where σ is the standard deviation of the elements of K^{CCT} , $\alpha > 0$ [23, 33].

3.4 Clustering Quality Evaluation

To evaluate the community detection performance, we employ the Adjusted Rand Index (ARI) introduced in [15]. Some advantages of this quality index are listed in [25].

ARI is based on the Rand Index (RI) introduced in [28]. The Rand Index quantifies the level of agreement between two partitions of n elements X and Y . Given a as the number of pairs of elements that are in the same clusters in both partitions, and b the number of pairs of elements in different clusters in both partitions, the Rand Index is defined as $\frac{a+b}{\binom{n}{2}}$.

The Adjusted Rand Index is the transformation of the Rand Index such that its expected value is 0 and maximum value is 1: $\text{ARI} = \frac{\text{Index} - \text{ExpectedIndex}}{\text{MaxIndex} - \text{ExpectedIndex}}$.

4 Proximity-Based Community Detection in Attributed Networks

In order to apply the proximity measures described in Sect. 3.3 to attributed networks, we need a way to embed node attribute information into the adjacency matrix. This can be done by modifying edge weights based on the attributes:

$$a_{ij}^s = \beta a_{ij} + (1 - \beta) s_{ij}, \quad (2)$$

where $\beta \in [0, 1]$ and $s_{ij} = s(\mathbf{f}_i, \mathbf{f}_j)$ is an attribute similarity measure calculated for nodes i and j . An attribute similarity measure, as the name implies, shows to what extent two nodes are similar by attributes.

By varying the coefficient β , we can make a trade-off between weighted adjacency and attribute similarity. So, when $\beta = 0$, the attributed adjacency matrix A^s describes only nodes similarity by attributes, while with $\beta = 1$ it coincides with A .

Given A^s , we can compute attributed versions of all the other matrices required to define proximity measures. Then, the proximity measures can be calculated and applied for detecting clusters using the Spectral method.

To take node attributes into account, we use various attribute similarity measures. Let $\mathbf{f}_i = (f_i^1, \dots, f_i^d)$ and $\mathbf{f}_j = (f_j^1, \dots, f_j^d)$ be the attribute vectors of nodes i and j , respectively. The attribute similarity measures are defined as following:

- Matching Coefficient³ [34]: $s^{\text{MC}}(\mathbf{f}_i, \mathbf{f}_j) = \frac{\sum_{k=1}^d \mathbb{1}(f_i^k = f_j^k)}{d}$, where $\mathbb{1}(x)$ is the indicator function which takes the value of one if the condition x is true and zero otherwise;
- Cosine Similarity [35, Chap. 2]: $s^{\text{CS}}(\mathbf{f}_i, \mathbf{f}_j) = \frac{\mathbf{f}_i \cdot \mathbf{f}_j}{\|\mathbf{f}_i\|_2 \|\mathbf{f}_j\|_2}$;
- Extended Jaccard Similarity [35, Chap. 2]: $s^{\text{JS}}(\mathbf{f}_i, \mathbf{f}_j) = \frac{\mathbf{f}_i \cdot \mathbf{f}_j}{\|\mathbf{f}_i\|_2^2 + \|\mathbf{f}_j\|_2^2 - \mathbf{f}_i \cdot \mathbf{f}_j}$;
- Manhattan Similarity [8]: $s^{\text{MS}}(\mathbf{f}_i, \mathbf{f}_j) = \frac{1}{1 + \|\mathbf{f}_i - \mathbf{f}_j\|_1}$;
- Euclidean Similarity [8]: $s^{\text{ES}}(\mathbf{f}_i, \mathbf{f}_j) = \frac{1}{1 + \|\mathbf{f}_i - \mathbf{f}_j\|_2}$.

5 Experiments

In this section, we compare attribute-aware proximity measures with the plain ones in experiments with several real-world datasets:

- WebKB [21]: a dataset of university web pages. Each web page is classified into one of five classes: course, faculty, student, project, staff. Each node is associated with a binary feature vector ($d = 1703$) describing presence or absence of words from the dictionary. This dataset consists of four unweighted graphs: Washington ($n = 230$, $m = 446$), Wisconsin ($n = 265$, $m = 530$), Cornell ($n = 195$, $m = 304$), and Texas ($n = 187$, $m = 328$).
- CiteSeer [30]: an unweighted citation graph of scientific papers. The dataset contains 3312 nodes and 4732 edges. Each paper in the graph is classified into one of six classes (the topic of the paper) and associated with a binary vector ($d = 3703$) describing the presence of words from the dictionary.
- Cora [30]: an unweighted citation graph of scientific papers with a structure similar to the CiteSeer graph. The number of nodes: $n = 2708$, the number of edges: $m = 5429$, the number of classes: $c = 7$, and the number of words in the dictionary (the length of the feature vector): $d = 1433$.

These datasets are clustered using multiple methods. First, we apply the k -means algorithm, which uses only attribute information and ignores graph structure. Then, each dataset is clustered with the Spectral algorithm and five plain proximity measures that do not use attribute information. Finally, communities are detected using the Spectral algorithm and attribute-aware proximity measures that employ both data dimensions (structure and attributes).

We use balanced versions of attribute similarity measures with $\beta = \frac{1}{2}$ in (2).

Each of the proximity measures depends on the parameter. So, we search for the optimal parameter in the experiments, and the results include clustering quality for the optimal parameter.

³ Since equality will be rare for continuous attributes, Matching Coefficient is mainly used for discrete attributes, especially binary ones.

6 Results

In this section, we discuss the results of the experiments.

In Table 1, ARI for all the tested proximity measures and similarity measures on all the datasets is presented. “No” column shows the result for plain proximity measures that do not use attribute information. The table also presents ARI for the k -means clustering algorithm. The top-performing similarity measure is marked in red for each proximity measure.

As we can see, taking attributes into account improves community detection quality for all the proximity measures. Attribute-aware proximity measures outperform k -means for all the datasets except Texas. Therefore, we can conclude that in most cases, the proximity measures based on structure and attribute information perform better than both the plain proximity measures which use only structure information and the k -means clustering method which uses only attribute information.

Not all tested attribute similarity measures have shown good clustering quality. Figure 1 presents the average rank and standard deviation for attribute similarity measures and k -means. The rank is averaged over 6 datasets. The figure contains 5 graphs: one for each proximity measure.

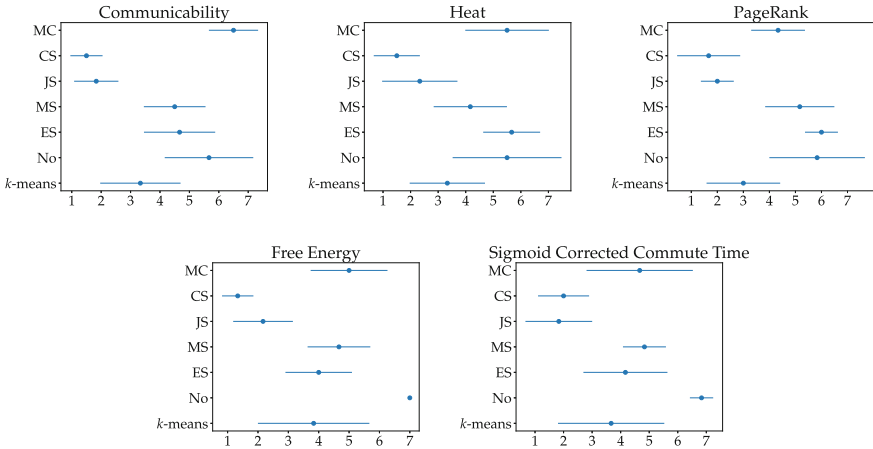


Fig. 1. Average rank and standard deviation for attribute similarity measures and k -means

One can see that the Cosine Similarity and Extended Jaccard Similarity measures perform the best: they have the highest ranks for all the proximity measures. As for plain proximity measures, which are not combined with any similarity measure, they have one of the lowest ranks. The performance of the Matching Coefficient, the Manhattan Similarity, and the Euclidean Similarity measures varies for different proximity measures.

Table 1. Results of the experiments

Prox. Measure	Similarity Measure					
	MC	CS	JS	MS	ES	No
Washington						
Communicability	0.048	0.458	0.352	0.13	0.24	0.05
Heat	0.058	0.457	0.444	0.093	0.055	0.043
PR	0.269	0.461	0.39	0.097	0.087	0.037
FE	0.291	0.461	0.322	0.129	0.243	0.009
SCCT	0.307	0.397	0.362	0.132	0.28	0.048
<i>k</i> -means	0.095					
Wisconsin						
Communicability	0.089	0.459	0.41	0.104	0.064	0.02
Heat	0.075	0.472	0.416	0.111	0.092	0.082
PR	0.126	0.471	0.36	0.13	0.067	0.045
FE	0.081	0.441	0.398	0.066	0.066	0.064
SCCT	0.056	0.354	0.383	0.103	0.089	0.045
<i>k</i> -means	0.364					
Cornell						
Communicability	0.012	0.2	0.107	0.034	0.058	0.035
Heat	0.064	0.181	0.109	0.069	0.046	0.072
PR	0.047	0.118	0.088	0.011	-0.025	0.063
FE	0.053	0.308	0.309	0.046	0.058	-0.013
SCCT	0.076	0.193	0.112	0.057	0.055	0.027
<i>k</i> -means	0.066					
Texas						
Communicability	0.118	0.288	0.281	0.177	0.23	0.008
Heat	0.137	0.212	0.289	0.076	0.156	-0.013
PR	0.221	0.174	0.287	0.073	0.133	0.0
FE	0.23	0.342	0.233	0.23	0.234	0.041
SCCT	0.274	0.344	0.258	0.21	0.253	0.15
<i>k</i> -means	0.409					
CiteSeer						
Communicability	0.0	0.24	0.282	0.001	0.001	0.113
Heat	0.001	0.258	0.276	0.005	0.004	0.112
PR	0.003	0.242	0.266	-0.0	0.001	0.109
FE	0.0	0.41	0.41	0.162	0.186	-0.001
SCCT	0.001	0.252	0.31	0.051	0.183	0.018
<i>k</i> -means	0.1					
Cora						
Communicability	0.0	0.107	0.119	0.09	0.027	0.002
Heat	0.071	0.083	0.061	0.076	0.024	0.002
PR	0.032	0.138	0.135	0.068	0.029	0.005
FE	0.023	0.408	0.404	0.301	0.236	-0.001
SCCT	0.025	0.156	0.189	0.127	0.184	0.002
<i>k</i> -means	0.07					

In Table 2, the top-performing combinations of a proximity measure and a similarity measure are presented. As can be seen, the undisputed leader is Free Energy combined with the Cosine Similarity measure.

Table 2. The top-performing pairs of proximity measure and similarity measure

Proximity measure	Similarity measure	Average rank
1 FE	CS	2.833
2 FE	JS	6.333
3 Communicability	CS	6.667
4 SCCT	JS	7.333
5 SCCT	CS	7.667
6 Communicability	JS	8.333
7 PR	CS	8.333
8 Heat	CS	8.667

7 Conclusion

In this paper, we investigated the possibility of applying proximity measures for community detection in attributed networks. We studied a number of proximity measures, including Communicability, Heat, PageRank, Free Energy, and Sigmoid Corrected Commute-Time. Attribute information was embedded into proximity measures using several attribute similarity measures, i.e., the Matching Coefficient, the Cosine Similarity, the Extended Jaccard Similarity, the Manhattan Similarity, and the Euclidean Similarity.

According to the results of the experiments, taking node attributes into account when measuring proximity improves the efficiency of proximity measures for community detection. Not all attribute similarity measures perform equally well. The top-performing attribute similarity measures were the Cosine Similarity and Extended Jaccard Similarity.

Future studies may address the problem of choosing the optimal β in (2). Another area for future research is to find more effective attribute similarity measures. Furthermore, the proposed method can be compared with the Embedding Approach of [20].

References

1. Avrachenkov, K., Chebotarev, P., Rubanov, D.: Kernels on graphs as proximity measures. In: International Workshop on Algorithms and Models for the Web-Graph. LNCS, vol. 10519, pp. 27–41. Springer (2017)
2. Aynulin, R.: Efficiency of transformations of proximity measures for graph clustering. In: International Workshop on Algorithms and Models for the Web-Graph. LNCS, vol. 11631, pp. 16–29. Springer (2019)

3. Bothorel, C., Cruz, J.D., Magnani, M., Micenkova, B.: Clustering attributed graphs: models, measures and methods. *Netw. Sci.* **3**(3), 408–444 (2015)
4. Chebotarev, P.Y., Shamis, E.: On the proximity measure for graph vertices provided by the inverse Laplacian characteristic matrix. In: 5th Conference of the International Linear Algebra Society, Georgia State University, Atlanta, pp. 30–31 (1995)
5. Chebotarev, P.: The walk distances in graphs. *Discrete Appl. Math.* **160**(10–11), 1484–1500 (2012)
6. Chunaev, P.: Community detection in node-attributed social networks: a survey. *Comput. Sci. Rev.* **37**, 100286 (2020)
7. Costa, L.D.F., Oliveira Jr., O.N., Travieso, G., Rodrigues, F.A., Villas Boas, P.R., Antiquiera, L., Viana, M.P., Correa Rocha, L.E.: Analyzing and modeling real-world phenomena with complex networks: a survey of applications. *Adv. Phys.* **60**(3), 329–412 (2011)
8. Dang, T., Viennet, E.: Community detection based on structural and attribute similarities. In: International Conference on Digital Society (ICDS), pp. 7–12 (2012)
9. Deza, M.M., Deza, E.: *Encyclopedia of Distances*, 4th edn. Springer, Berlin (2016)
10. Dijkstra, E.W., et al.: A note on two problems in connexion with graphs. *Numerische Mathematik* **1**(1), 269–271 (1959)
11. Estrada, E.: The communicability distance in graphs. *Linear Algebra Appl.* **436**(11), 4317–4328 (2012)
12. Fouss, F., Francoisse, K., Yen, L., Pirotte, A., Saerens, M.: An experimental investigation of kernels on graphs for collaborative recommendation and semisupervised classification. *Neural Netw.* **31**, 53–72 (2012)
13. Fouss, F., Yen, L., Pirotte, A., Saerens, M.: An experimental investigation of graph kernels on a collaborative recommendation task. In: Sixth International Conference on Data Mining (ICDM'06), pp. 863–868. IEEE (2006)
14. Girvan, M., Newman, M.E.: Community structure in social and biological networks. *Proc. Nat. Acad. Sci.* **99**(12), 7821–7826 (2002)
15. Hubert, L., Arabie, P.: Comparing partitions. *J. Classif.* **2**(1), 193–218 (1985)
16. Ivashkin, V., Chebotarev, P.: Do logarithmic proximity measures outperform plain ones in graph clustering? In: International Conference on Network Analysis. PROMS, vol. 197, pp. 87–105. Springer (2016)
17. Jain, A.K.: Data clustering: 50 years beyond k -means. *Pattern Recogn. Lett.* **31**(8), 651–666 (2010)
18. Kivimäki, I., Shimbo, M., Saerens, M.: Developments in the theory of randomized shortest paths with a comparison of graph node distances. *Physica A Stat. Mech. Appl.* **393**, 600–616 (2014)
19. Kondor, R., Lafferty, J.: Diffusion kernels on graphs and other discrete input spaces. In: International Conference on Machine Learning, pp. 315–322 (2002)
20. Li, Y., Sha, C., Huang, X., Zhang, Y.: Community detection in attributed graphs: an embedding approach. In: Thirty-Second AAAI Conference on Artificial Intelligence, pp. 338–345 (2018)
21. Lu, Q., Getoor, L.: Link-based classification. In: Proceedings of the 20th International Conference on Machine Learning (ICML-03), pp. 496–503 (2003)
22. von Luxburg, U.: A tutorial on spectral clustering. *Stat. Comput.* **17**(4), 395–416 (2007)
23. von Luxburg, U., Radl, A., Hein, M.: Getting lost in space: large sample analysis of the resistance distance. In: Advances in Neural Information Processing Systems, pp. 2622–2630 (2010)

24. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297. Oakland, CA, USA (1967)
25. Milligan, G.W., Cooper, M.C.: A study of the comparability of external criteria for hierarchical cluster analysis. *Multivar. Behav. Res.* **21**(4), 441–458 (1986)
26. Neville, J., Adler, M., Jensen, D.: Clustering relational data using attribute and link information. In: Proceedings of the Text Mining and Link Analysis Workshop, 18th International Joint Conference on Artificial Intelligence, pp. 9–15 (2003)
27. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. Technical report, Stanford InfoLab (1999)
28. Rand, W.M.: Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **66**(336), 846–850 (1971)
29. Ruan, Y., Fuhry, D., Parthasarathy, S.: Efficient community detection in large networks using content and links. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 1089–1098 (2013)
30. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. *AI Mag.* **29**(3), 93 (2008)
31. Sharpe, G.: Solution of the $(m+1)$ -terminal resistive network problem by means of metric geometry. In: Proceedings of the First Asilomar Conference on Circuits and Systems, Pacific Grove, CA, pp. 319–328 (1967)
32. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905 (2000)
33. Sommer, F., Fouss, F., Saerens, M.: Comparison of graph node distances on clustering tasks. In: International Conference on Artificial Neural Networks. LNCS, vol. 9886, pp. 192–201. Springer (2016)
34. Sulc, Z., Řezanková, H.: Evaluation of recent similarity measures for categorical data. In: Proceedings of the 17th International Conference Applications of Mathematics and Statistics in Economics. Wydawnictwo Uniwersytetu Ekonomicznego we Wrocławiu, Wrocław, pp. 249–258 (2014)
35. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson Education India (2016)
36. Yang, J., McAuley, J., Leskovec, J.: Community detection in networks with node attributes. In: 2013 IEEE 13th International Conference on Data Mining, pp. 1151–1156. IEEE (2013)
37. Zhou, Y., Cheng, H., Yu, J.X.: Graph clustering based on structural/attribute similarities. *Proc. VLDB Endow.* **2**(1), 718–729 (2009)



Core Method for Community Detection

A. A. Chepovskiy¹ (✉), S. P. Khaykova², and D. A. Leshchev³

¹ Department of Applied Mathematics, HSE Tikhonov Moscow Institute of Electronics and Mathematics (MIEM HSE), Myasnienskaya Street 20, Moscow 101000, Russian Federation
c4hapa@gmail.com

² MIEM HSE, Moscow, Russia

³ Skolkovo Institute of Science and Technology (Skoltech), Moscow, Russia

Abstract. The processing of networks of interacting objects makes it possible to solve topical issues in the modern world of identifying opinion leaders and channels for the dissemination and exchange of information. In this work, the structure of networks of interacting objects and their possible analysis with the help of weighted graphs based on the interaction of elements of such networks are considered. At the beginning of the work, a methodology for working with a weighted graph was proposed. It is called by the authors the “core method” and provides an algorithm for the analyst’s actions to identify communication groups, opinion leaders and disseminate information in the network. The key concepts of the γ -core of the graph, the interaction coefficients and the density of communities and the core are introduced. In the second part of the work, the main capabilities of the software developed by the authors, which allows the operator to carry out the procedures required for the method, visualize the results and export the obtained data, are presented. The third part shows the application of the “core method” on a weighted graph, based on the data about the coverage of the activities of the Moscow city authorities in the fight against the new coronavirus infection Covid-19 imported from Twitter. This example shows how opinion leaders on a weighted graph can be identified using the core method and the implemented application.

Keywords: Social network analysis · Algorithms for network analysis · Networks visual representation · Community detection

1 Theory

1.1 About Revealing Communities and Key Applied Tasks

The problem of identifying implicit communities in networks of interacting objects has been covered in many publications over the past 20 years [1–7]. Various algorithms have been developed to solve problems related to this topic. Separately, it is possible to highlight the currently relevant area of analysis of social networks [8–10]. When working with social networks graphs, the key applied problems are the following:

1. the task of determining the proximity of user profiles, the coincidence of their interests, the degree of (face-to-face) acquaintance;

2. the task of recognizing opinion leaders;
3. the task of identifying channels for the distribution and exchange of information between users.

To solve these problems, one must first decide what kind of graph should be built when importing data from the original source. There is usually no point in working with the graph of the entire social network due to its large size. Therefore, as a rule, a subgraph, the construction of which is carried out using a breadth-first search from a set of vertices given in advance, is unloaded. We will work with weighted graphs $G(V, E)$, the set of vertices V of which consists of the original objects – users of the social network. In this case, the weight on the set of edges E is given by the function w with nonnegative values and corresponds to the degree of intensity of interaction of objects with each other. The method for determining the values of $w(e)$ depends on the specific source network. The weight $w(v)$ of the vertex v is then defined as the sum of the weights of all edges incident to it. And let the weight of a given set of vertices be defined as the sum of the weights of all these vertices. The concept of a community is defined in many works [11–15]. Community S is a subgraph (containing set of vertices), with the density of edges between them higher than in the whole graph. In this work, we assume that the communities do not overlap, i.e. after the selection of communities, each vertex is in a single community. Therefore, the community weight $w(S)$ is determined accordingly.

We also define the concept of the internal weight of a community $w^*(S)$ – the sum of the weights of the edges, both vertices of which lie inside the community and the internal weight of the vertex $w^*(v)$ – the sum of the weights of the edges incident to it that lie in the community of the given vertex. For the tasks listed above, you can build the corresponding weighted graphs: graphs of general similarity of users; graphs of user sympathy; graphs of information interaction of users.

To solve the first task and build a graph of general similarity of users, even an unweighted graph of mutual friends or subscriptions is often sufficient. When solving this problem, a weighted graph, the values of the weights of the edges of which are determined based on the general attributes of the original network, can also be used [16]. To basically solve Task 2, both the user sympathy graph and the graph of their information interaction can be used. These weighted graphs are based on the values of the network attributes. But for a qualitative analysis of the graph of information interaction, it is necessary to perform several procedures, which are described further in our work. This allows you to qualitatively solve tasks 2 and 3.

The high-quality construction of the specified graphs requires the selection of parameters for the weight of the edges, depending on the initial network of object interaction. For example, when importing data from the Twitter network, it is possible to use information about existing likes, retweets, comments, user subscriptions. These types of interactions will constitute many attributes. In this case, one of the options for constructing weights on the edges between two vertices is to calculate a weighted sum for these vertices based on the attributes.

1.2 Removing “Garbage” Vertices and Allocating the Core

If we consider the original graph $G(V, E)$ and apply popular methods of community revealing to it, then the picture will usually be distorted by a large number of leaf vertices obtained during data import. Other vertices, for which the weight of incident edges is significantly lower than the others in the graph, are also possible. Typically, these will be vertices of users who minimally interact with the rest. For graphs of information interaction, these users are not interesting. We will conditionally call such vertices as “garbage”. In contrast, the key vertices of opinion leaders, which have a lot of weight, as well as the structure of communities, including the “heaviest” of them. Revealing such vertices is one of the important tasks, because around them “heavy” communities are formed, and other vertices are attracted. It will be more accurate to say that a vertex v will be called δ -garbage if its weight is less than δ . Then the set $Junk_\delta(G)$ of all garbage in the graph is defined as follows:

$$Junk_\delta(G) = \{v \in V | w(v) < \delta\} \quad (1)$$

Mirror situation takes place for vertices with large weight value. Let us call α -star or simply a “star” such vertices v of the graph G , that v has a weight greater than some value α . The set of stars $Star_\alpha(G)$ is then defined as follows:

$$Star_\alpha(G) = \{v \in V | \omega(v) > \alpha\} \quad (2)$$

Such vertices attract other to their communities, unless they, in turn, are in communities with a significant total weight. In this case, the weight of the edges between adjacent vertices from different communities will be important [17]. One of the goals of the core method is to select the key core community (or several such communities) that has the greatest weight among the other communities in graph G . Further, for a given partition, we denote the community with the maximum weight by $Core(G)$, and its weight by $w(Core(G))$. It may happen that in the initial graph several communities S_i are revealed, whose weight is very close or even coincides with the maximum weight $w(Core(G))$. In these cases, we can talk about the presence of several communities-cores in the graph G . The admissible degree of proximity of these values is denoted by γ . We define the γ -core $Core_\gamma(G)$ as the set of vertices from those communities that satisfy the following relation:

$$Core_\gamma(G) = \left\{ v \in V | v \in S_i : \frac{w(S_i)}{w(Core(G))} > \gamma \right\} \quad (3)$$

In addition to cores and stars, the graph often contains other smaller communities, the connection inside which is quite dense, and the weight is high enough that cores and other large communities cannot absorb these smaller communities.

1.3 Graphs of Information Interaction

It is possible to simplify the graph analysis task by ignoring weak interactions of the original objects by removing garbage vertices. This can be done in two ways. The first

is simply by removing garbage vertices and then the remaining from them edges. But we will use another method: for all edges having a weight less than a given β we will consider this weight equal to zero, i.e. remove such edges and obtain a new set of edges E' . After that, some of the vertices will become isolated and can be removed from the analyzed graph. Thus we get a new set of vertices V' . Let us denote the graph obtained after these operations as $G'(V', E')$ – the graph of active information interaction of network objects.

The second method is better suited for forming such a graph, because by taking β equal to δ , we can remove not only $Junk_\delta(G)$, but other inactive edges and vertices from the graph.

We define the interaction coefficient $k_{int}(G')$ as the ratio of the doubled number of edges to the square of the number of vertices in the resulting graph of active information interaction G' :

$$k_{int}(G') = \frac{2|E'|}{|V'|^2} \quad (4)$$

It is easy to see that in the complete graph $k_{int}(G') = 1 - \frac{1}{n}$, which for large n is close to 1. But graphs of networks of interacting objects are sparse, so usually this coefficient will be closer to 0. High values of $k_{int}(G')$ will indicate a significant connection between actively interacting vertices. This indicator is generally responsible for the activity within the analyzed graph. Similarly, you can determine the coefficients of interaction within individual communities of the graph. For the core case, high values will indicate a high level of subjectivity of the graph content [17].

Let us define $k_S(G')$ – the density coefficient of the community S as the ratio of the total internal weight of the vertices of this community to the doubled weight of the graph edges, which is equal to the ratio of the weight of the edges within the community to the weight of the edges of the graph:

$$k_S(G') = \frac{\sum_{v \in S} w^*(v)}{2 \sum_{e \in S} w(e)} = \frac{\sum_{e \in S} w(e)}{\sum_{e \in E'} w(e)} \quad (5)$$

Similarly we define $k_{Core_\gamma}(G')$ – the density coefficient of the γ -core as the ratio of the total weight of the vertices of the γ -core $Core_\gamma(G')$ to the doubled total weight of the edges of the graph G' :

$$k_{Core_\gamma}(G') = \frac{\sum_{v \in Core_\gamma(G')} w(v)}{2 \sum_{e \in E'} w(e)} \quad (6)$$

High values of $k_{Core_\gamma}(G')$ show, that the core in such a graph plays a significant role in comparison with other communities and garbage vertices. Based on these coefficients, the classification and methodology for working with graphs will be built.

1.4 Meta-vertices and Meta-graph

Sometimes, to analyze the graph of interacting objects, it may be interesting to consider communities as meta-vertices and work with this new meta-graph (we will denote such

a meta-graph as $G_{,1}$ for the first iteration). Then, with each iteration, the number of vertices will decrease: each meta-vertex in the new graph is a group of vertices of the previous graph. Consequently, the meta-vertex is itself a certain graph, within which it is useful to apply the algorithm and analyze the selected communities.

After revealing communities in the meta-graph and analyzing their connections with each other, we get new meta-vertices. Repeating the operation of forming meta-vertices in this way, we obtain the graph $G_{,2}$. Similarly, at the i -th step, the graph $G_{,i}$ is obtained. Thus, at one of the iterations, you can get a general view of the interaction of the largest groups of users. Of course, this makes sense for a large initial graph G .

1.5 Core Method

Let us present the sequence of analysts' actions to solve the main tasks 2 and 3. It is this technique that we will call the "core method". Acting according to the algorithm presented below, the analyst will be able to identify both opinion leaders and ways of disseminating information within and between communities.

1. **Remove isolated vertices.** The presence of such may be due to the peculiarities of the source network and the data import process. We get the graph G .
2. **Calculate the initial interaction coefficient of the graph $k_{int}(G)$.** This will be important as a reference point for a graph without isolated vertices.
3. **Remove garbage vertices.** To do this, it is necessary to determine the value of β for which to perform the operation of removing edges. We get the graph G' .
4. **Calculate the updated interaction coefficient of the graph $k_{int}(G')$.** The recommended variation range is within the following limits: $0, 8 < \frac{k_{int}(G')}{k_{int}(G)} < 0, 9$. In case the coefficient has changed outside this range, we recommend returning to step 3 and making it with a different value of β .
5. **Apply algorithm to reveal communities.** It is supposed to use an algorithm that identifies not overlapping communities. For example, variations of the algorithms Infomap [18, 19], Louvain [20] etc. can be used.
6. **Identify stars.** Choose the value of α and select the set of vertices $Star_{\alpha}(G)$, consisting of the stars of the graph. Check that stars are highlighted in the largest communities and such communities have a high $k_{S_i}(G')$. If not, then change the α .
7. **Detect the core.** Determine γ and compose $Core_{\gamma}(G')$.
8. **Generate meta-graph.** Create a graph of meta-vertices $G_{,1}$. Reveal communities inside $Core_{\gamma}(G)$ and other key meta-vertices, study their structure in accordance with the initial tasks of the researcher. If necessary, continue working with each meta-vertex separately, passing for them recursively to step 2.
9. **Create a meta-meta graph.** Consider the structure of $G_{,2}$ in accordance with the original tasks.

2 Tool

The authors of this work have designed and implemented a special tool for graph analysis that supports automatic revealing of communities through built-in algorithms and visualization of the result, as well as other actions that allow implementing the previously indicated methodology.

Since the graph of interacting objects can be obtained from any social network and other sources, it was extremely important to fix a universal graph representation format that allows storing the attributes of its vertices and edges. A special XML-like unified markup format called AVS is used. The description of the vertices and edges of the graph is written to the AVS format file: their names, attribute names, data format in attributes. The description of the graph is followed by the description of each vertex and each edge.

An important feature of this AVS-format is its way of storing the attributes of the edges and vertices of the graph, which are large texts. In the AVS file itself, only the link to the file (or part of the file) is stored in the field of the corresponding attribute. And the text is stored in the file, which allows you to analyze texts separately, as well as perform data compression if necessary. Therefore, when exporting data from the developed application, the graph itself (vertices, edges and their attributes) is saved in AVS file, and user texts are stored in separate yaml files for each of them.

The developed software allows user to implement the following user scenarios:

- load an AVS file with a graph for visualization and point-by-point consideration of its vertices and connections;
- apply to the loaded graph any of algorithms to reveal communities available in the application and get acquainted with the statistics of the resulting partition;
- analyze partitions, including using meta-vertices, in order to change communities and / or weight functions;
- export the resulting set of communities for analysis or presentation of results in other systems or manually.

The graph uploaded by the user is displayed on the scene of the main application window (Fig. 1). Scene – is an area of the main application window used to display the graph and interact with its components.

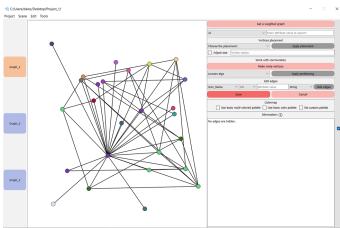


Fig. 1. Application screen and the scene

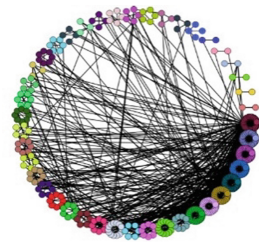


Fig. 2. Revealed communities, graph G'

Dividing the graph into communities is one of the most essential tools for their analysis. In the implemented application, two algorithms to reveal communities are integrated: Infomap and Louvain.

After applying the partition, the vertices of the graph belonging to the same community will be colored in the same color, each vertex will be added an attribute with its community number. Then vertices are arranged by revealed communities (Fig. 2).

In addition to these basic actions, this tool allows you to perform those that are important for presented methodology. One can remove edges with a predicate weight, remove isolated vertices, compose meta-vertices, define stars and select the core (using the statistics menu). To determine β and remove “garbage” vertices, it is possible to hide the edges of the graph on the scene, according to the user-specified condition, and later either remove them completely from the graph or cancel the hiding and return the hidden edges to the scene. This functionality provides analysts with the ability to visually evaluate the graph that will be obtained for different β and choose its optimal value. The user can specify a condition for hiding edges not only by weight, but also by any other attribute of the edge, if there are any in the graph G . To do this, you must specify the edge attribute, sign and constant value for comparison, as well as the type of comparison. The tool also provides the ability to remove isolated vertices, which is especially useful after removing edges, since, with the optimal choice of β , some of the vertices will become isolated and their removal will help to finally clear the graph of “garbage”. To create a meta-graph, the “Make meta-vertices” button is implemented, which initiates the creation, based on the current scene, of meta-vertices corresponding to the communities selected at the current step.

To create a graph of meta-vertices, first, for each community, the total weight of edges (with both vertices within the same community) is calculated. This gives the weight of the meta-vertices used to calculate the radius for displaying on the main scene. Then, for each pair of communities, the total weight of the edges between the vertices of these communities is calculated. This is how the weights of the new edges in the meta-graph are determined.

When analyzing a meta-graph, it may be relevant for a user to study the structure of a community representing one meta-vertex. To do this, the user can “fall” into the meta-vertex and see the subgraph of the original graph, composed only of the vertices of the given community. With this view of the subgraph, the user can save and open the subgraph as a separate project and work with it using the full functionality of the application. The statistics menu implemented in the application allows the user to study the topological indicators of the graph and its components (vertices, communities). Basic statistics, located in the text boxes in the lower left corner, will help user to calculate the graph interaction ratio needed in the early stages of the core method, as well as the community and γ -core density factors for later stages. Communities, sorted by the number of vertices included in them (the value is indicated in parentheses after the name of the community), will help user to quickly find the largest one and highlight the core.

3 An Example of Applying the Method on Data from Twitter

3.1 The Core Detection

As an example, in the evening of 05/27/2020, 8 relevant posts, regarding the actions of the Moscow city authorities in the fight against the new coronavirus infection Covid-19, were downloaded from Twitter. As well as related comments, likes, retweets. Among the posts were both the official statements of the city leadership in the media and their accounts, and highly social messages of a provocative nature from opposition-minded individuals. The download took place with the generation of a weighted graph based on interaction with the original posts, as well as other actions previously performed by users. Based on each of the interactions of users with each other (subscription, like, comment, retweet), the weight of the edges between them was formed.

First, a graph with 632 vertices and 1002 edges was obtained. Further, acting according to the previously described core method, the following actions were performed. First, isolated vertices were removed, and 459 vertices remained with 1002 edges, this will be graph G . Indicators of graph G : mean edges weight = 2.767, mean vertex degree = 1.585, max vertex degree = 126, $k_{int}(G) = 0,0095$. Then we go to step 3 and “remove garbage vertices”: choosing the value $\beta = 1$, remove 249 edges with weights not exceeding β and 34 vertices (which became isolated after removing edges), we get the graph G' . We calculate the coefficient $k_{int}(G') = 0,0083$. The changes can be estimated as follows: $\frac{k_{int}(G')}{k_{int}(G)} = 0,87$, which is in the recommended range. The total weight of the edges: $\sum_{e \in E'} w(e) = 2773$.

Next, we apply the Infomap algorithm to the graph G' to reveal implicit communities. We get 43 communities (Fig. 2), 8 of which contain more than 15 vertices, and calculate the main indicators (Table 1). Four communities: S_0, S_1, S_2 and S_4 have both a high density coefficient and a high maximum internal degree. This indicates the presence of stars and active interaction within these communities. Communities S_5 and S_7 may also contain stars, while communities S_3 and S_6 rather do not have stars in their composition. However, the density of S_3 community is high.

Let's look at the vertices of the graph G' with the maximum weights (Table 2). Mean weighted vertex degree in G' is 12,08. The last column, obtained as the weight of the vertex divided by this value, shows well the stars-vertices with a given indicator above 14. Therefore, we will take $\alpha = 170 > 12,08 * 14 = 169,12$.

Thus, we have found $Star_\alpha(G)$, and for $\alpha = 170$ this set consists of 5 vertices. The community S_0 has the greatest weight, so we take $Core(G) = S_0$, and define $\gamma = 0,77$. Then, according to (3) $Core_\gamma(G')$ contains S_0, S_1, S_2 and S_4 . We calculate $k_{Core_\gamma}(G') = \frac{2195}{2 \times 2773} = 0,404$. A high value obtained indicates a correctly found core.

Now we can generate the meta-graph $G'_{,1}$. Further, inside the key meta-vertices $G'_{,1}$, including those from the $Core_\gamma(G')$, you can look at their structure from the inside.

3.2 The Structure of Meta-Vertices

Consider the partition into internal communities in S_1 (Fig. 3). This meta-vertex is a source of information – a star-vertex corresponding to the official media account and its

Table 1. Communities with at least 15 nodes

S_i	$ S_i $	$w(S_i)$	$w^*(S_i)$	$k_{S_i}(G')$	$\max_{v \in S_i} w(v)$	$\max_{v \in S_i} w^*(v)$
S_0	44	710	382	0,068	445	187
S_1	49	576	408	0,073	315	204
S_2	33	537	296	0,053	258	129
S_3	25	444	288	0,051	60	30
S_4	44	372	306	0,055	171	128
S_5	24	340	170	0,03	171	83
S_6	24	337	180	0,032	64	27
S_7	22	167	134	0,024	84	61

Table 2. Vertices of G' with top weight

Encrypted vertex nick	Vertex degree	Weighted vertex degree	Weighted vertex degree in community	Weighted degree divided by mean
v_***ov	126	445	187	37,08
le***al	84	315	204	26,25
Vl***va	76	258	129	21,5
Mr***ay	65	171	128	14,25
Pr***at	62	171	83	14,25
Ol***13	27	84	61	7
aa***an	38	64	27	5,33
8a***Wn	20	60	30	5
ma***n_	25	57	29	4,75
Se***us	20	49	23	4,08
ru***60	31	42	15	3,5
NπA***36	13	41	18	3,41
dj***ef	12	40	15	3,33

adjacent vertices. We will call such meta-vertices “constellations of the first kind”, and “planets” – its adjacent vertices. Thus, in our classification, S_1 is a constellation of the first kind, consisting of one vertex-star and 48 vertices-planets.

Let us consider in more detail the meta-vertex corresponding to $S_0 \in Core_\gamma(G)$. Let’s reveal internal communities in S_0 using Infomap algorithm (Fig. 4). The top-star is

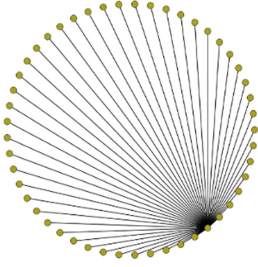


Fig. 3. Internal structure of community S_1

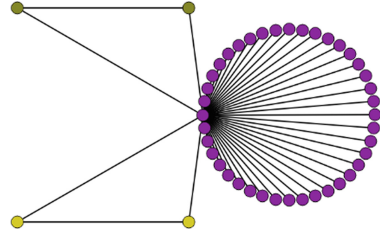


Fig. 4. Internal structure of community S_0

clearly visible – this is a pronounced influencer and the adjacent users who had interaction with the original posts. It is worth noting that the composition of many of these users is ambiguous: basic nicknames of 15 random characters, which are created upon registration, the number of followers is zero or close to zero, photos are uploaded mostly without a face. Presumably, such users are bots or fakes of the respective influencer. Of course, there are also real users who share the leader’s views. They can even form their own communities, in this case there are two of them, but both of them consist of 2 vertices. Further there will be examples where additional communities are larger.

We will call such meta-vertices “constellations of the second kind”, “stars” in them – opinion leaders, and “planets” – other vertices (in general, not all of them will be adjacent to a star). Thus, in our classification S_0 is a constellation of the second kind, consisting of one vertex-star, with which 43 vertex-planets are connected.

Communities S_2 and S_4 also represent constellations of the second kind (Figs. 5, 6) consisting of one star, 32 and 43 planetary vertices, respectively. Other vertices, according to the value of α are not stars here.

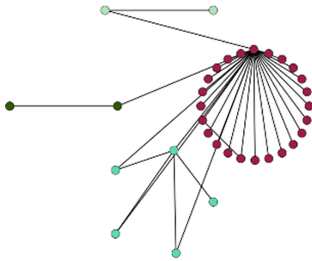


Fig. 5. Internal structure of community S_2

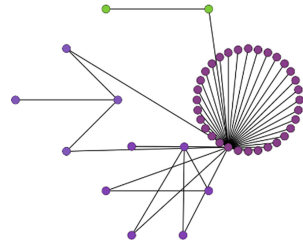


Fig. 6. Internal structure of community S_4

Community S_3 is a “constellation of the third kind” (Fig. 7), there is no star here, but the density value $k_{S_3}(G')$ is quite high, the vertices are still competing for supremacy in this group. This means that with the further development of the social network over time, a star will appear here.

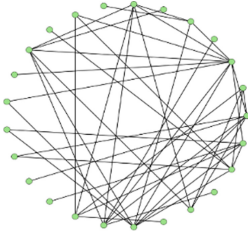


Fig. 7. Internal structure of community S_3

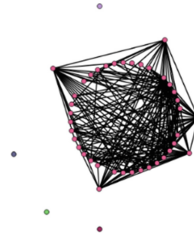


Fig. 8. G'_2 – the graph of meta-meta vertices with revealed communities

These constellation variants are not unique for the considered communities and can be found in other cases as well, for example, in the structure of communities S_5 , S_6 and S_7 . There will also be one of the three previous pictures. Among them, the star is only S_5 for the taken value of α .

Thus, this technique distinguishes opinion leaders and communities, constituting support groups for the respective leaders. It should be noted that the qualitative analysis identified leaders of various opinions, both pro-government (Fig. 9) and opposition (Fig. 10).



Fig. 9. Star vertex @Vi***va



Fig. 10. Star vertex @Mr***ay

Then we repeat the selection of the community in the graph of meta-vertices G'_1 and obtain the graph of meta-meta-vertices G'_2 . If the graph G'_1 can be conventionally called a “constellation graph”, then the graph G'_2 – is a “galaxy graph” (Fig. 8). Five communities are distinguished on G'_2 . Only one of them includes more than 1 meta-vertex. It consists of 39 meta-vertices and is the “Galactic core” for the initial graph G' . All the other 4 are composed of single meta-vertex, each of which, in turn, consists of several vertices of the original graph. It should be noted that there could be more interesting “graphs of galaxies”.

4 Conclusions

This paper describes a core method that allows you to analyze weighted graphs and solve the problem of identifying opinion leaders and ways of disseminating information. Mathematical indicators for a weighted graph, which make it possible to assess the degree of interaction of network vertices, have been introduced. The functionality of the software implemented by the authors, which allows analysts to work within the framework of the suggested method, is described. An example of work by the method using the described application with a graph built on the basis of real data from the Twitter network is given and considered in detail.

The authors see a possible further development in this area in testing the hypothesis of self-organization of networks of interacting objects in the formation of implicit communities in the process of graph evolution based on interaction according to laws similar to the laws of physics and with the aim of bringing the system into a state of stable equilibrium.

References

1. Aggarwal, C.: *Social Network Data Analytics*. Springer, US (2011)
2. Chepovskiy, A., Lobanova, S.: Combined method to detect communities in graphs of interacting objects. *Bus. Inf.* **42**(4), 64–73 (2017)
3. Coscia, M., Rossetti, G., Giannotti, F., Pedreschi, D.: Demon: a local-first discovery method for overlapping communities. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 615–623, ACM (2012)
4. Karimi, F., Lotfi, S., Izadkhah, H.: Multiplex community detection in complex networks using an evolutionary approach. *Expert Syst. Appl.* **146**, 113184 (2020)
5. Lambiotte, R., Rosvall, M.: Ranking and clustering of nodes in networks with smart teleportation. *Phys. Rev. E* **85**, 056107 (2012)
6. Newman, M., Girvan, M.: Finding and evaluating community structure in networks. *Phys. Rev. E* **69**(2), 1–15 (2004)
7. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**, 814–818 (2005)
8. Lei, T., Huan, L.: Community detection and mining in social media. *Synthesis Lectures on Data Mining and Knowledge Discovery*, p. 137 (2010)
9. Xie, J., Szymanski, B.K., Liu, X.: SLPA: uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In: *2011 IEEE 11th International Conference, Data Mining Workshops (ICDM)*, pp. 344–349 (2011)
10. Yang Bo, Liu Dayou, Liu Jiming.: *Discovering communities from social networks: methodologies and applications*. Handbook of Social Network Technologies and Applications, pp. 331–346. Springer, Boston (2010)
11. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. *Phys. Rev. E* **70**, 066111 (2004)
12. Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**(3), 75–174 (2010)
13. Girvan, M., Newman, M.: Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* **99**(12), 7821–7826 (2002)
14. Lancichinetti, A., Fortunato, S.: Community detection algorithms: a comparative analysis. *Phys. Rev. E* **80**, 056117 (2009)

15. Radicchi, F., Castellano, C., Loreto, V., Cecconi, F., Parisi, D.: Defining and identifying communities in networks. *Proc. Natl. Acad. Sci. U.S.A.* **101**(9), 2658–2663 (2004)
16. Leschyov, D.A., Suchkov, D.V., Khaykova, S.P., Chepovskiy, A.A.: Algorithms to reveal communication groups. *Voprosy kiberbezopasnosti.* **32**(4), 61–71 (2019). <https://doi.org/10.21681/2311-3456-2019-4-61-71>
17. Voronin, A.N., Kovaleva, J.B., Chepovskiy, A.A.: Interconnection of network characteristics and subjectivity of network communities in the social network Twitter. *Voprosy kiberbezopasnosti.* **37**(3), 40–57 (2020). <https://doi.org/10.21681/2311-3456-2020-03-40-57>
18. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. U.S.A.* **105**(4), 1118–1123 (2008)
19. Rosvall, M., Bergstrom, C.T., Axelsson, D.: The map equation. *Eur. Phys. J. Spec. Top.* **178**(1), 13–23 (2009)
20. Blondel, V., Guillaume, J., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large network. *J. Stat. Mech. Theory Exp.* **10**, P10008 (2008)



Effects of Community Structure in Social Networks on Speed of Information Diffusion

Nako Tsuda and Sho Tsugawa^(✉)

University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan
n.tsuda@mibel.cs.tsukuba.ac.jp, s-tugawa@cs.tsukuba.ac.jp

Abstract. Social media users can widely disseminate information, potentially affecting societal trends. Therefore, understanding factors affecting information diffusion in social media is important for successful viral marketing campaigns. In this paper, we focus on the community structure of social networks among Twitter users and investigate how that structure affects the speed of diffusion by retweets. Extracting communities among sampled Twitter users, we investigate differences in diffusion speed between tweets with many intra-community retweets and those with many inter-community retweets. Consequently, we show that tweets with many intra-community retweets tend to spread slowly. We also use community structures in Twitter social networks to tackle the tasks of predicting time intervals between a first tweet and its N -th retweet. We show the potential of community structure features in a social network for predicting information diffusion speed.

Keywords: Social network · Community structure · Diffusion speed · Social media

1 Introduction

Social media users can widely disseminate information, potentially affecting societal trends [2]. When information about a given product is widely disseminated, users receiving that information may purchase that product, increasing its sales. Therefore, understanding factors affecting information diffusion in social media is important for successful viral marketing campaigns.

In our previous work, we have investigated how community structures of users in social media affect the scale of cascading information diffusion [15, 16]. Many social networks have a community structure, in which the network is composed of highly clustered communities with sparse links between them [4, 12]. Our previous studies have shown that if information is spread across different communities, the information will be widely spread [15, 16]. Other existing studies have also reported that the community structure has strong influence on spreading processes using real data and theoretical models [3, 6, 7, 10, 11, 13, 21].

While many previous studies have focused on the *scale* of information diffusion cascades [1, 8, 9, 15, 16, 21], there have been few studies of factors affecting its *speed*, which is equally important. For instance, suppose a post reaches ten thousand users within one week, and another reaches ten thousand users within one hour. Both posts have the same diffusion scale, but significantly different diffusion speeds. For viral marketing campaigns, it is important to spread information both widely and rapidly among social media users. Therefore, it is also important to understand the factors affecting diffusion speed of information cascades.

In this paper, we focus on the community structure of social networks among Twitter users and investigate how that structure affects the speed of diffusion by retweets. Extracting communities among sampled Twitter users, we investigate differences in diffusion speed between tweets with many intra-community retweets and those with many inter-community retweets. We also use community structures in Twitter social networks to tackle the tasks of predicting the diffusion speed of a given tweet (i.e., time intervals between a first tweet and its N -th retweet). Consequently, we examine the effectiveness of community structure features in a social network for predicting information diffusion speed. Our main contributions are summarized as follows.

- We investigate the effects of inter-community and intra-community diffusion of tweets on their diffusion speed. To the best of our knowledge, this is the first study to investigate the effects of community structure of a social network on diffusion speed of tweets.
- We show the potential of community structure features for predicting information diffusion speed. While many studies have addressed the tasks of predicting the scale of diffusion, it has been rarely studied the tasks of predicting diffusion speed. Our results contribute to constructing models for predicting the speed of information diffusion.

The remainder of this paper is organized as follows. In Sect. 2, we investigate the effects of community structure on diffusion speed of tweets. In Sect. 3, we tackle the tasks of predicting diffusion speed of tweets. Finally, Sect. 4 contains our conclusions and a discussion of future work.

2 Effects of Community Structure on Diffusion Speed of Tweets

2.1 Methodology

The following analyses use the same user set as in our previous work [16]. We have been collecting data regarding followers and followees of 356,453 Twitter users every month. From the collected data, we have constructed user social networks as of early January 2016. In the constructed social network, each user is represented as a node, and a directed link (u, v) exists if user u follows user v . For this study, we examined these users' retweets during January 2016, representing 1,626,183 original tweets and 5,496,832 retweets.

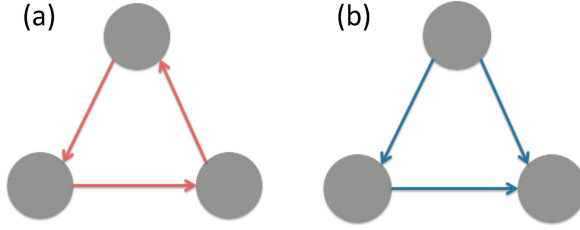


Fig. 1. (a) cycle triangle (b) flow triangle

Among several community detection algorithms [5, 17], we used the community extraction algorithm proposed in [14] to the constructed social network to extract cycle and flow k -truss communities following our previous study [15]. The algorithm for extracting the cycle truss and flow truss can control the strength of community structure by tuning the parameter k . We extract community structures with different strengths and analyze how the strength of community structure affects the extent of tweet diffusion. The cycle k -truss community and the flow k -truss community are defined as generalization of k -truss [20] in undirected networks to directed networks. The three-node relationships shown in Fig. 1a and b are defined as cycle triangle, and flow triangle, respectively. Cycle k -truss communities are extracted as follows. We first obtain all cycle triangles in the given network G . Then, for each link (i, j) , we count the number of cycle triangles associated with link (i, j) , denoted as $c(i, j)$. We finally remove all links with $c(i, j) < k$ from the network G . Each connected component with at least two nodes of the remaining network is a cycle k -truss community. Note that singleton nodes do not belong to any communities. By replacing “cycle” in the above description with “flow”, the flow truss community can be similarly extracted. The truss number k is a nonnegative integer value of $0 \leq k \leq d_{max} - 1$, where d_{max} is the maximum node degree in the network. As k increases, the extracted k -truss communities only contain nodes with a larger number of shared triangles. Therefore, when k is large, strongly clustered and small k -truss communities are obtained whereas when k is small, weakly clustered and large communities are obtained. In this paper, we regard communities extracted with larger value of k as stronger communities than communities extracted with smaller value of k .

As a measure of tweet diffusion speed, we obtained the N th retweet time of a tweet by calculating the time interval between the original tweet and its N th retweet [18, 19]. We investigate the relation between N th retweet time and the intra-community diffusion rate among the first N retweets of tweet t . The intra-community diffusion rate among the first N retweets of tweet t is the ratio of users who are in the same community with the user who posts original tweet t among users who post retweets of tweet t . More specifically, it is defined as

$$p_t(N) = \frac{|\{u | u \in U_{N,t} \cap c(u) = c(u(t))\}|}{|U_{N,t}|}, \quad (1)$$

where $U_{N,t}$ is the set of target users who post the first through N -th retweets of tweet t , $c(u)$ is the community to which user u belongs, and $u(t)$ is the user who posted the original tweet t . By changing N , we investigated the relation between N th retweet time and the intra-community diffusion rate $p(N)$.

2.2 Results

We categorized tweets into four groups based on the intra-community diffusion rate $p(N)$: tweets with $0 \leq p < 0.25$, $0.25 \leq p < 0.5$, $0.5 \leq p < 0.75$, and $0.75 \leq p \leq 1$. We then compared average N th retweet times among the four groups (Fig. 2). The following shows the results for cycle truss communities with $k = 5$ and flow truss communities with $k = 15$. Note that cycle truss communities with $k = 5$ and flow truss communities with $k = 15$ have similar community sizes.

For both cycle truss communities and flow truss communities, Fig. 2 shows that the higher the intra-community diffusion rate, the longer it takes to reach N times. For example, it takes approximately 1.9 times longer to reach the 100th retweet when the intra-community diffusion rate p is $0.75 \leq p \leq 1$ than when $0 \leq p < 0.25$. We also confirmed similar tendencies with k values other than those shown in Fig. 2. These results suggest that the intra-community diffusion rate affects the speed of tweet diffusion.

We next extracted truss communities with different community structure strengths and performed the same analysis. Previous analyses have shown that tweet diffusion times can be long when the intra-community diffusion rate is high. Therefore, we next categorized tweets into four patterns based on the intra-community diffusion rate before N th retweets, and for each investigated the extent to which retweet times differ between the cases of diffusion within relatively strong and weak community structures. Figure 3 compares the N th retweet time in extracted $k = 5$, $k = 10$, and $k = 20$ cycle truss communities with intra-community diffusion rate p values where $0 \leq p < 0.25$ (Fig. 3a), $0.25 \leq p < 0.50$ (Fig. 3b), $0.50 \leq p < 0.75$ (Fig. 3c), and $0.75 \leq p \leq 1$ (Fig. 3d). Figure 4 shows similar results for extracted flow truss communities with $k = 10$, $k = 30$, and $k = 50$.

Figures 3 and 4 do not show large differences when intra-community diffusion rates are low, but when the rate is 0.25 or more, tweet diffusion takes longer when the community structure is strong than when it is weak. In a cycle truss community where intra-community diffusion rate p is $0.25 \leq p < 0.50$, for example, it takes approximately 1.8 times longer to reach 100 retweets when $k = 20$ than when $k = 5$. These results suggest that when the intra-community diffusion rate is at least some level, stronger community structures incur longer diffusion times than do weak structures.

3 Predicting Diffusion Speed

3.1 Problem Setting

In this section, we tackled prediction tasks similar to those in a previous study [1]. Specifically, we predict the N th retweet time, when a tweet has been retweeted

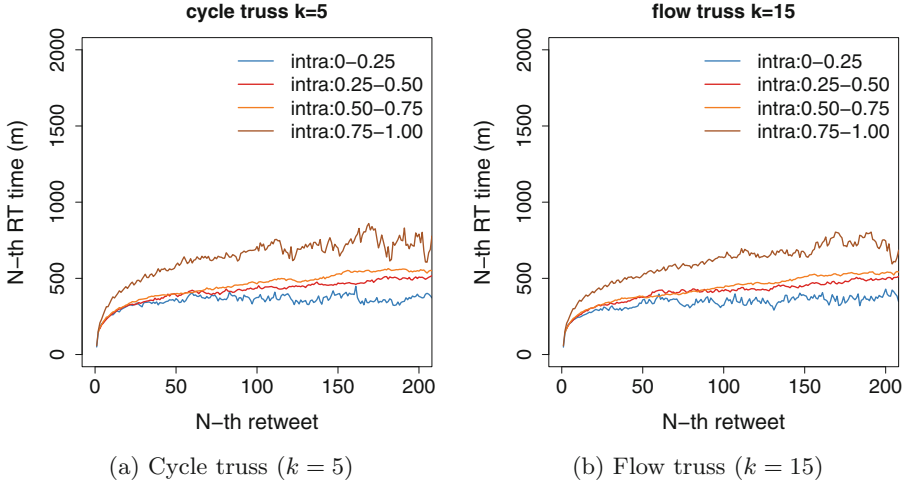


Fig. 2. Comparison of N th retweet times among four groupings based on intra-community diffusion rate.

100 times and was first retweeted more than sixty minutes after its initial posting. We used tweets posted within a certain period and retweets of those posts as training data. From these data, we extracted features regarding the tweet body texts, the reliability and activity of the tweeting or retweeting user, and community structure features for predictions.

The target users were the same 356,453 users analyzed in the previous Section. The training period was the period from January 1st, 2016 to April 30th, 2016. The testing period was the period from May 1st, 2016 to May 20th, 2016. We only use tweets that are retweeted 100 times more, and their lifetimes are sixty minutes and more. Thus, 8,438 original tweets were available for the training data, and 1,489 original tweets were available for the test data.

3.2 Prediction Method

We predict the N th retweet time based on the method proposed in [1], because it has been shown to be useful for predicting both diffusion scales and lifetimes, which is a type of diffusion time. In this method, features necessary for predictions are extracted from the training data, and stored in a knowledge base. Given a test-data tweet subject to predictions, we extract tweets having similar features with the test tweet from the knowledge base (i.e., training data), calculate their average of the N th retweet time, and use the result as the predicted value.

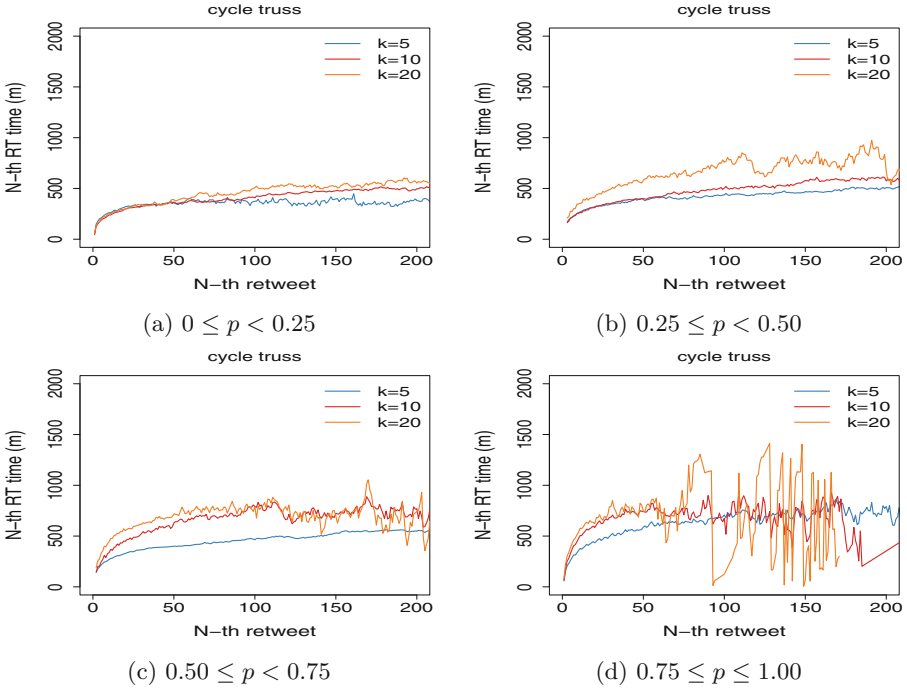


Fig. 3. Comparison of N th retweet times among different truss numbers k (cycle truss).

To verify the usefulness of community structure features, we evaluated and compared the results of predicting the N th retweet time under each of four prediction methods: the existing method proposed in [1], a method combining the existing method and one using community structure features, a method using only community structure features, and a method using the average of all baseline learning data serving as prediction values.

We first describe the existing method [1]. In this method, based on the approach similar to k -nearest neighbor algorithm, tweets similar to the target for prediction are extracted from a knowledge base, and their average diffusion time is used as the predicted value. The problem here is how to measure similarity between tweets. As features for measuring similarity, this method uses following features.

- Frequency of character bigrams in the tweet body
- Reliability $R(u)$ of the person initially posting the tweet
- Activity $A(u)$ of the person initially posting the tweet
- Information amount in the initial tweet
- Time to retweet interval (TRI)
- Number of retweets posted per timeslot (RTI).

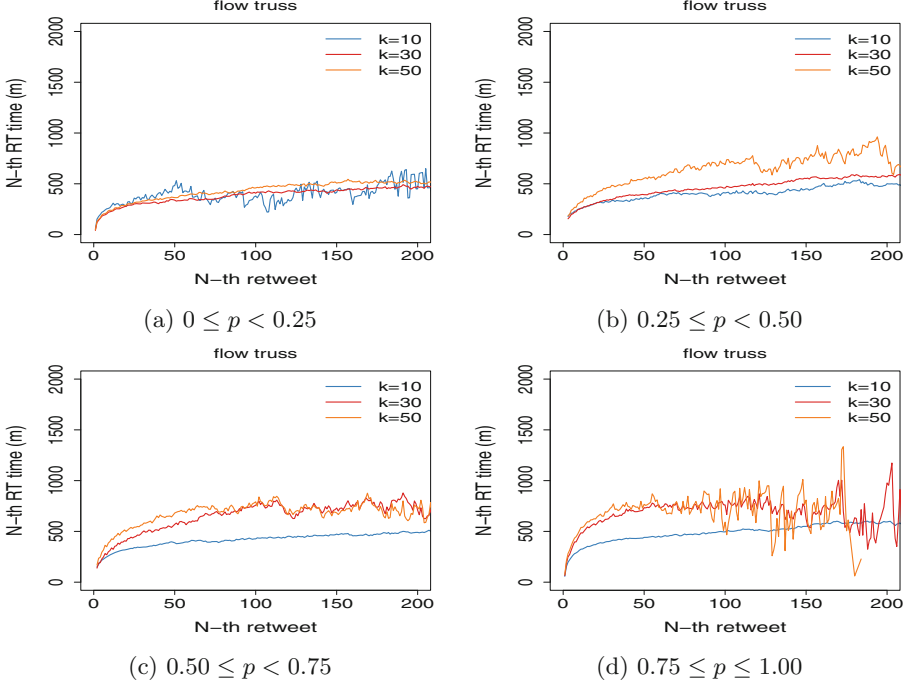


Fig. 4. Comparison of N th retweet times among different truss numbers k (flow truss).

See Ref. [1] for details.

Given a tweet that is the target for prediction, we obtain Jaccard coefficients from character bigrams in tweet texts in the knowledge base and text of the target tweet, and extract α tweets having the highest Jaccard coefficients. We further calculate Euclidean distances between TRI and RTI for the prediction tweet and the top α tweets and extract the top β tweets with nearest distance. From the prediction tweet (et_i), reliability (R) of the top β tweets (ht_i), activity (A), and information amount (I), we calculate

$$DIST(et_i, ht_i) = \sqrt{[R(et_i) - R(ht_i)]^2 + [A(et_i) - A(ht_i)]^2 + [I(et_i) - I(ht_i)]^2}$$

and extract the top γ tweets with smallest $DIST$. Finally, we calculate the average of the N th retweet times for the γ tweets with shortest distance between the prediction tweet (et_i) and the prediction target, taking this as the prediction value. Following Ref. [9], in this study we used values $\alpha = 50$, $\beta = 10$, and $\gamma = 5$.

We next describe the method for prediction by combining community structure features with the existing method. As a community structure feature, we use the intra-community diffusion rate $p(100)$, calculated as described in Sect. 2. By using intra-community diffusion rate $p(100)$ and features used in the existing method, we measure the similarity between the prediction target tweet and those

in the knowledge base. Note that the method for extracting communities is the same as that described in Sect. 2. Given a tweet that is subject to prediction, we extract from the knowledge base tweets with similar intra-community diffusion rates. Specifically, we calculate the quotient obtained by dividing $p(100)$ of the prediction target tweet by 0.05. We also calculate the quotients obtained by dividing $p(100)$ for knowledge base tweets by 0.05, and extract knowledge base tweets with the same quotients as the prediction tweet. After that, we perform the same processing for predictions as in the existing method, and extract similar tweets from the knowledge base, and calculate the average of their N th retweet time.

For the prediction method using only community structure features, we only use the intra-community diffusion rate $p(100)$ for measuring the similarity between the target tweet and tweets in the knowledge base. Given a tweet that is subject to prediction, we calculate the quotients obtained by dividing $p(100)$ for knowledge base tweets and those subject to prediction by 0.05, extract knowledge base tweets with same quotients as the prediction tweet, and use the average of the N th retweet times for the extracted tweets as the prediction value.

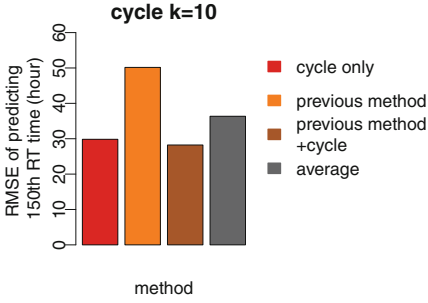
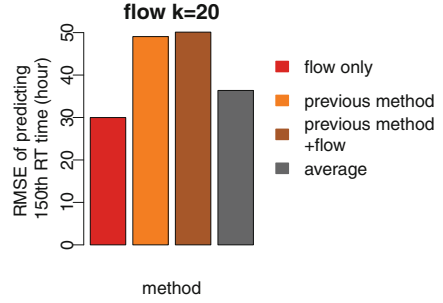
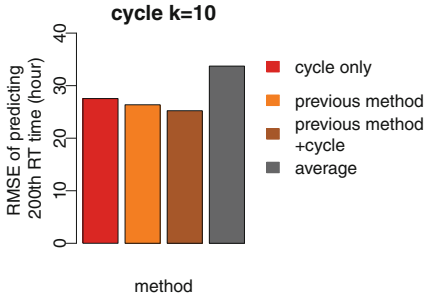
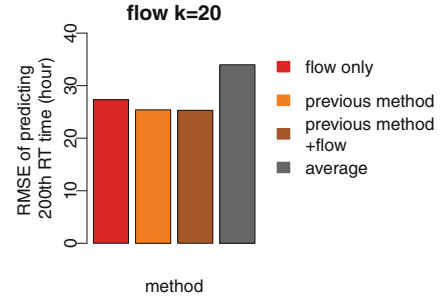
As a baseline, we also use the method of using the average of all training data as the predicted value. In this method, for a given target tweet, the prediction value is the average of the N th retweet times in the training data.

3.3 Prediction Results

We performed experiments for predicting N th retweet times under settings $N = 150$ and $N = 200$. First, we show differences in accuracy when predicting N th retweet times under the existing method, the method using only community features, and the method combining the conventional method with the community features method. For comparison, we also performed predictions using the baseline method. Figures 5 and 6 respectively show root-mean-square error (RMSE) as a measure of prediction accuracy for each method in the cases where $N = 150$ and $N = 200$. Note that lower RMSE means higher prediction accuracy. As community structure features, we show the results of using $k = 10$ for cycle truss communities and $k = 20$ for flow truss communities.

Figure 5 shows that when predicting 150th retweet times, the method using only community structure features has a smaller RMSE than does the existing and baseline methods. This suggests the usefulness of community structure features for predicting the N th retweet time. However, when combining the existing and community structure feature methods, RMSE becomes a large value for flow truss communities with $k = 20$. There is thus a need for further investigation of methods combining community structure features with other features.

Figure 6 shows that when predicting 200th retweet times, the combination of the existing and community structure features methods gives the most accurate predictions. However, there is not a large difference in RMSE between that case and when using the existing method.

(a) Cycle truss ($k = 10$)(b) Flow truss ($k = 20$)**Fig. 5.** RMSE for 150th retweet time predictions.(a) Cycle truss ($k = 10$)(b) Flow truss ($k = 20$)**Fig. 6.** RMSE for 200th retweet time predictions.

Overall, these results show the potential of community structure features for predicting N th retweet times. In contrast, it is also shown that there remains room for consideration regarding the use of those features and their combination with other features.

4 Conclusion

In this paper, we have investigated how the community structure of a social network among Twitter users affects the speed of diffusion by retweets. Extracting communities among sampled Twitter users, we investigate differences in diffusion speed between tweets with many intra-community retweets and those with many inter-community retweets. Consequently, we have shown that tweets with many intra-community retweets tend to spread slowly. We have also tackled the tasks of predicting time intervals between a first tweet and its N th retweet using features obtained from community structures in Twitter social networks. Our

results have shown the potential of community structure features that are effective for predicting information diffusion speed. In contrast, we have also found that there remains room for consideration regarding the use of those features and their combination with other features for predicting diffusion speed.

In future work, generalizability of the results should be investigated. The results in this paper should be validated using other datasets. Although clear differences between cycle and flow truss communities are not observed in this paper, analyzing the effects of different types of communities on information diffusion is an important future work. We are also interested in understanding the background mechanisms of the effects of community structure on diffusion speed.

References

1. Bae, Y., Ryu, P.M., Kim, H.: Predicting the lifespan and retweet times of tweets based on multiple feature analysis. *ETRI J.* **36**(3), 418–428 (2014)
2. Bakshy, E., Hofman, J.M., Mason, W.A., Watts, D.J.: Everyone’s an influencer: quantifying influence on Twitter. In: *Proceedings of WSDM 2011*, pp. 65–74 (2011)
3. De Meo, P., Ferrara, E., Fiumara, G., Provetti, A.: On Facebook, most ties are weak. *Commun. ACM* **57**(11), 78–84 (2014)
4. Ferrara, E.: A large-scale community structure analysis in Facebook. *EPJ Data Sci.* **1**(1), 9 (2012)
5. Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**(3), 75–174 (2010)
6. Galstyan, A., Cohen, P.: Cascading dynamics in modular networks. *Phys. Rev. E* **75**(3), 036109 (2007)
7. He, J.L., Fu, Y., Chen, D.B.: A novel top-k strategy for influence maximization in complex networks with community structure. *PLoS ONE* **10**(12), e0145283 (2015)
8. Hong, L., Dan, O., Davison, B.D.: Predicting popular messages in Twitter. In: *Proceedings of the 20th International Conference Companion on World Wide Web*, pp. 57–58 (2011)
9. Kong, S., Feng, L., Sun, G., Luo, K.: Predicting lifespans of popular tweets in microblog. In: *Proceedings of SIGIR 2012*, pp. 1129–1130 (2012)
10. Li, C.T., Lin, Y.J., Yeh, M.Y.: The roles of network communities in social information diffusion. In: *Proceedings of the IEEE Big Data 2015*, pp. 391–400 (2015)
11. Nematzadeh, A., Ferrara, E., Flammini, A., Ahn, Y.Y.: Optimal network modularity for information diffusion. *Phys. Rev. Lett.* **113**(8), 088701 (2014)
12. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. *Phys. Rev. E* **69**(2), 026113 (2004)
13. Onnela, J.P., Saramäki, J., Hyvönen, J., Szabó, G., Lazer, D., Kaski, K., Kertész, J., Barabási, A.L.: Structure and tie strengths in mobile communication networks. *PNAS* **104**(18), 7332–7336 (2007)
14. Takaguchi, T., Yoshida, Y.: Cycle and flow trusses in directed networks. *R. Soc. Open Sci.* **3**(11), 160270 (2016)
15. Tsuda, N., Tsugawa, S.: Effects of truss structure of social network on information diffusion among Twitter users. In: *Proceedings of INCoS 2019*, pp. 306–315 (2019)
16. Tsugawa, S.: Empirical analysis of the relation between community structure and cascading retweet diffusion. In: *Proceedings of ICWSM 2019* (2019)
17. Tsugawa, S.: A survey of social network analysis techniques and their applications to socially aware networking. *IEICE Trans. Commun.* **102**(1), 17–39 (2019)

18. Tsugawa, S., Ohsaki, H.: Negative messages spread rapidly and widely on social media. In: Proceedings of COSN 2015, pp. 151–160 (2015)
19. Tsugawa, S., Ohsaki, H.: On the relation between message sentiment and its virality on social media. *Soc. Netw. Anal. Min.* **7**(1), 19:1–19:14 (2017)
20. Wang, J., Cheng, J.: Truss decomposition in massive networks. *Proc. VLDB Endow.* **5**(9), 812–823 (2012)
21. Weng, L., Menczer, F., Ahn, Y.Y.: Virality prediction and community structure in social networks. *Sci. Rep.* **3**, 2522 (2013)



Closure Coefficient in Complex Directed Networks

Mingshan Jia^(✉), Bogdan Gabrys, and Katarzyna Musiał

University of Technology Sydney, Ultimo, NSW 2007, Australia
mingshan.jia@student.uts.edu.au,
{bogdan.gabrys,katarzyna.musial-gabrys}@uts.edu.au

Abstract. The 3-clique formation, a natural phenomenon in real-world networks, is typically measured by the local clustering coefficient, where the focal node serves as the centre-node in an open triad. The local closure coefficient provides a novel perspective, with the focal node serving as the end-node. It has shown some interesting properties in network analysis, yet it cannot be applied to complex directed networks. Here, we propose the *directed closure coefficient* as an extension of the closure coefficient in directed networks, and we extend it to weighted and signed networks. In order to better use it in network analysis, we introduce further the *source closure coefficient* and the *target closure coefficient*. Our experiments show that the proposed directed closure coefficient provides complementary information to the classic directed clustering coefficient. We also demonstrate that adding closure coefficients leads to better performance in link prediction task in most directed networks.

Keywords: Clustering coefficient · Closure coefficient · Directed networks

1 Introduction

Networks, abstracting the interactions between components, are fundamental in studying complex systems in a variety of domains ranging from cellular and neural networks to social, communication and trade networks [1, 2]. Small subgraph patterns (also known as motifs [3]) that recur at a higher frequency than those in random networks are crucial in understanding and analysing networks. Motifs underlie many descriptive and predictive applications such as community detection [4, 5], anomaly detection [6, 7], role analysis [8, 9], and link prediction [10].

Among them, 3-node connected subgraphs, which are the building blocks for higher-order motifs, are explored most often. Further, the 3-clique, or the triadic closure from a temporal perspective, has been revealed to be a natural phenomenon of networks across different areas [3, 11]. Nodes sharing a common neighbour are more likely to connect with each other. For example, in an undirected friendship network, there is an increased likelihood for two people having a common friend to become friends [12]; in a directed citation network, a paper

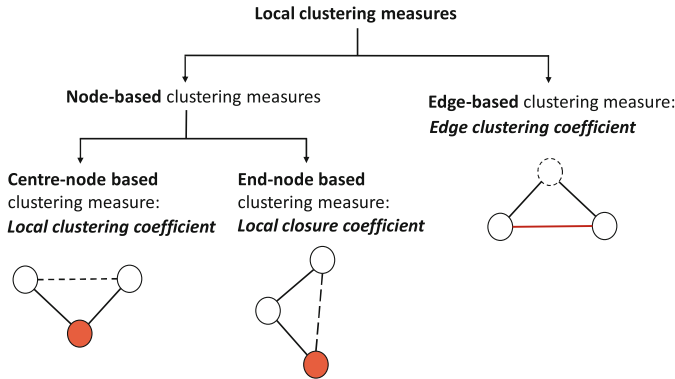


Fig. 1. Classification diagram of local clustering measures. In each of the two node-based clustering measures, the focal node is painted in red, and the dotted edge represents the potential closing edge in an open triad. In the edge-based clustering measure, the focal edge is in red, and the dotted outline circle represents the potential node that forms a triangle.

cites two papers where one tends to cite the other [13]; and in a signed directed trust network, when Alice distrusts Bob, Alice discounts anything recommended by Bob [14].

The classic measure of a 3-clique formation is the *local clustering coefficient* [15], which is defined by the percentage of the number of triangles formed with a node (referred to as node i) to the number of triangles that i could possibly form with its neighbours. In this definition, the focal node i serves as the centre-node in an open triad. To emphasize, an open triad is an unordered pair of edges sharing one node. With a focus on node i , it describes the extent to which edges congregate around it. The extensions of local clustering coefficient have been thoroughly discussed for weighted networks [16, 17], directed networks [18] and signed networks [19]. Another metric for 3-clique formation, with a focus on an edge, is the *edge clustering coefficient* [20] which evaluates to what extent nodes cluster around this edge.

A recent study has proposed another local edge clustering measure, i.e., the *local closure coefficient* [21]. With the focal node i as the end-node of an open triad, it is quantified as the percentage of two times the number of triangles containing i to the number of open triads with i as the end-node. Conceptually, the local clustering coefficient measures the phenomenon that two friends of mine are also friends themselves, while the local closure coefficient is focusing on a friend of my friend is also a friend of mine. This new metric has been proven to be a useful tool in several network analysis tasks such as community detection and link prediction [21]. Together with the two measures mentioned above, we propose a classification diagram of all three local clustering measures (Fig. 1).

The local closure coefficient is originally defined for undirected binary networks. However, in real-world complex networks, the relationships between components can be nonreciprocal (a follower is often not followed back by the followee), heterogeneous (trade volumes between countries vary significantly), and negative (an individual can be disliked or distrusted).

In this paper, with an end-node focus, we propose the *local directed closure coefficient* to measure local edge clustering in binary directed networks, and we extend it to weighted directed networks and weighted signed directed networks. Since in a directed 3-clique, each of the three edges can take either direction, there are eight different triangles in total. According to the direction of the closing edge, i.e., the edge that closes an open triad and forms a triangle, we classify them into two groups (emanating from or pointing to the focal node, as shown in Fig. 2). Based on that, we propose the *source closure coefficient* and the *target closure coefficient* respectively.

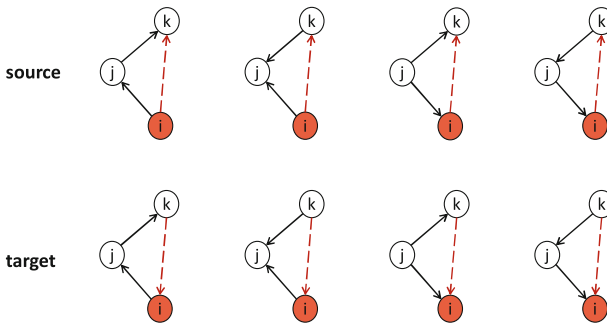


Fig. 2. Taxonomy of directed triangles. Two solid edges connecting nodes i , j and k form an open triad, which is closed by a dotted edge connecting nodes i and k . Focal node i , painted in red, is the end-node of an open triad. Eight triangles are classified into two groups according to the direction of the closing edge. First row shows a group where the focal node serves as the source node of the closing edge; second row is another group where the focal node serves as the target.

Our evaluations have revealed some interesting properties of the proposed metric. Through a correlation analysis on various networks, it is shown that the directed closure coefficient provides complementary information to the classical metric, i.e., the directed clustering coefficient. In a link prediction task, we propose two indices that include the source closure coefficient and the target closure coefficient. We show that in most networks, adding closure coefficients leads to better performance.

In summary, we propose (1) the local directed closure coefficient as another measure of edge clustering in directed networks; (2) an extension of it to weighted and signed networks; and (3) the source closure coefficient and the target closure coefficient. Through multiple experiments, we exhibit intrinsic features of the proposed metrics and how they can be used to improve certain network analysis tasks.

2 Preliminaries

This section introduces the preliminary knowledge of our work, including the classic clustering coefficient and the recently proposed closure coefficient.

2.1 Clustering Coefficient

The notion of local clustering coefficient was originally proposed bearing the name clustering coefficient, in order to measure the cliquishness of a neighbourhood in an undirected graph [15].

Let $G = (V, E)$ be an undirected graph on a node set V (the number of nodes is $|V|$) and an edge set E , without multiple edges and self-loops. The adjacency matrix of G is denoted as $\mathbf{A} = \{a_{ij}\}$. $a_{ij} = 1$ if there is an edge between node i and node j , otherwise $a_{ij} = 0$. We denote the degree of node i as $d_i = \sum_j a_{ij}$.

For any node $i \in V$, the *local clustering coefficient* is calculated as the number of triangles formed with node i and its neighbours (labelled as $T(i)$), divided by the number of open triads with i as the centre-node (labelled as $OT_c(i)$):

$$C_c(i) = \frac{T(i)}{OT_c(i)} = \frac{\frac{1}{2} \sum_j \sum_k a_{ij} a_{ik} a_{jk}}{\frac{1}{2} d_i (d_i - 1)}. \tag{1}$$

The subscript c here emphasizes that the focal node i serves as the centre-node of an open triad. We assume that $C_c(i)$ is well defined ($d_i > 1$). Clearly, $C_c(i) \in [0, 1]$.

In order to measure clustering at the network-level, the *average clustering coefficient* is introduced by averaging the local clustering coefficient over all nodes (an undefined local clustering coefficient is treated as zero): $\overline{C_c} = \frac{1}{|V|} \sum_{i \in V} C_c(i)$.

2.2 Closure Coefficient

Recently Yin et al. [21] proposed the *local closure coefficient*. Different from the ordinary centre-node focus in the local clustering coefficient, this definition is based on the end-node of an open triad. Recall that an open triad is an unordered pair of edges sharing one node. For example, in an open triad ijk with two edges ij and jk , there is no difference between (ij, jk) and (jk, ij) .

The local closure coefficient of node i is defined as two times the number of triangles formed with i (labelled as $T(i)$), divided by the number of open triads with i as the end-node. (labelled as $OT_e(i)$):

$$C_e(i) = \frac{2T(i)}{OT_e(i)} = \frac{\sum_j \sum_k a_{ij} a_{ik} a_{jk}}{\sum_{j \in N(i)} (d_j - 1)}, \tag{2}$$

where $N(i)$ denotes the set of neighbours of node i . $C_e(i)$ is well defined when the neighbours of i are not solely connected to it. $T(i)$ is multiplied by two for the reason that each triangle contains two open triads with i as the end-node.

When a triangle is actually formed (e.g., with nodes i , j and k), the focal node i can be viewed as the centre-node in one open triad (jik) or as the end-node in two open triads (ijk and ikj). Obviously, $C_e(i) \in [0, 1]$.

At the network-level, the *average closure coefficient* is then defined as the mean of the local closure coefficient over all nodes: $\overline{C_e} = \frac{1}{|V|} \sum_{i \in V} C_e(i)$. When we consider a random network where each pair of nodes is connected with probability p , its expected value is also p , i.e., $\mathbb{E}[\overline{C_e}] = p$.

3 Closure Coefficient in Directed Networks

In this section, we provide a general extension of the closure coefficient to directed networks, i.e., the local directed closure coefficient. We further propose the source and target closure coefficients. Finally, we extend it to weighted and signed networks.

3.1 Closure Coefficient in Binary Directed Networks

Motivated by the closure coefficient and the directed clustering coefficient [18], we aim to measure the directed 3-clique formation from the end-node of an open triad. There are eight different directed triangles, and a triangle (or an open triad) with bidirectional edges is treated as a combination of triangles (or open triads) with only unidirectional edges.

Let $\mathbf{A} = \{a_{ij}\}$ denote the adjacency matrix of a directed graph $G^D = (V, E)$. $a_{ij} = 1$ if there is an edge from node i to node j , otherwise $a_{ij} = 0$. The degree of node i is denoted as d_i , including both outgoing edges and incoming edges: $d_i = d_i^{out} + d_i^{in} = \sum_j a_{ij} + \sum_j a_{ji}$. The set of neighbours of node i is denoted $N(i)$. We now give the definition of the closure coefficient in directed networks.

Definition 1. *The local directed closure coefficient of node i in a directed network, denoted $C_e^D(i)$, is defined as twice the number of directed triangles formed with node i (labelled as $T^D(i)$), divided by twice the number of directed open triads with i as the end-node (labelled as $OT_e^D(i)$):*

$$C_e^D(i) = \frac{2T^D(i)}{2OT_e^D(i)} = \frac{\sum_j \sum_k (a_{ij} + a_{ji})(a_{ik} + a_{ki})(a_{jk} + a_{kj})}{2 \sum_{j \in N(i)} (a_{ij} + a_{ji})(d_j - (a_{ij} + a_{ji}))}. \quad (3)$$

$T^D(i)$ is multiplied by two since each triangle contains two open triads with i as the end-node. $OT_e^D(i)$ is multiplied by two because the closing edge of a directed open triad can take two directions. Obviously, $C_e^D(i) \in [0, 1]$. When the adjacency matrix \mathbf{A} is symmetric (the network becomes undirected), Eq. 3 reduces to Eq. 2, i.e., $C_e^D(i) = C_e(i)$.

Similarly, in order to measure at the network-level, we propose the definition of an average directed closure coefficient.

Definition 2. *The **average directed closure coefficient** of a directed network, denoted $\overline{C_e^{\mathcal{D}}}$, is defined as the average of the local directed closure coefficient over all nodes:*

$$\overline{C_e^{\mathcal{D}}} = \frac{1}{|V|} \sum_{i \in V} C_e^{\mathcal{D}}(i). \quad (4)$$

In a random network, where each directed edge occurs with a probability p , we also have $\mathbb{E}[C_e^{\mathcal{D}}(i)] = p$.

3.2 Closure Coefficients of Particular Patterns

In addition to a general measure, we propose to classify directed triangles into two groups according to the direction of the closing edge: one group where the focal node serves as the source node of the closing edge, another group where the focal node serves as the target (Fig. 2). Two definitions are given accordingly.

Definition 3. *For a given node i in a directed network, the **source closure coefficient**, denoted $C_e^{src}(i)$, and the **target closure coefficient**, denoted $C_e^{tgt}(i)$ are defined as:*

$$C_e^{src}(i) = \frac{\sum_j \sum_k (a_{ij} + a_{ji}) (a_{jk} + a_{kj}) a_{ik}}{2 \sum_{j \in N(i)} (a_{ij} + a_{ji}) (d_j - (a_{ij} + a_{ji}))},$$

$$C_e^{tgt}(i) = \frac{\sum_j \sum_k (a_{ij} + a_{ji}) (a_{jk} + a_{kj}) a_{ki}}{2 \sum_{j \in N(i)} (a_{ij} + a_{ji}) (d_j - (a_{ij} + a_{ji}))}.$$

Please note that $C_e^{src}(i) + C_e^{tgt}(i) = C_e^{\mathcal{D}}(i)$. These two metrics evaluate the extent to which the focal node is acting as the source node or the target node of the closing edges in a triangle formation. In Sect. 4.2, we show how the source and target closure coefficients can be used to improve the performance in a link prediction task.

3.3 Closure Coefficient in Weighted Networks

So far, the study is focusing on binary networks, where the value of every edge is either 1 or 0. In many networks, however, we need a more accurate representation of the relationships between nodes, such as the frequency of contact in a social network, the traffic flow in a road network, etc. Therefore we are interested in extending the closure coefficient for weighted networks.

In a weighted graph $G^{\mathcal{W}}$ described by its weight matrix $\mathbf{W} = \{w_{ij}\}$, we suppose $w_{ij} \in [0, 1]$ (normalised by the maximum weight), and the strength of node i is $s_i = \sum_j w_{ij}$. We introduce the weighted closure coefficient.

Definition 4. *The **weighted closure coefficient** of node i in a weighted network, denoted $C_e^{\mathcal{W}}(i)$, is defined as:*

$$C_e^{\mathcal{W}}(i) = \frac{\sum_j \sum_k w_{ij} w_{ik} w_{jk}}{\sum_{j \in N(i)} w_{ij} (s_j - w_{ij})}. \quad (5)$$

Obviously, $C_e^{\mathcal{W}}(i) \in [0, 1]$. When the weight matrix becomes binary, Eq. 5 degrades to Eq. 2, i.e., $C_e^{\mathcal{W}}(i) = C_e(i)$.

In a similar approach, the definition of closure coefficient in weighted directed networks can be extended from Eq. 3. Let us denote $\mathbf{W} = \{w_{ij}\}$ as the weight matrix of a weighted directed graph $G^{\mathcal{W}, \mathcal{D}}$, $w_{ij} \in [0, 1]$. The strength of node i is denoted by s_i ($s_i = \sum_j w_{ij} + \sum_j w_{ji}$).

Definition 5. *The weighted directed closure coefficient of node i , denoted $C_e^{\mathcal{W}, \mathcal{D}}(i)$, is defined as:*

$$C_e^{\mathcal{W}, \mathcal{D}}(i) = \frac{\sum_j \sum_k (w_{ij} + w_{ji})(w_{ik} + w_{ki})(w_{jk} + w_{kj})}{2 \sum_{j \in N(i)} (w_{ij} + w_{ji})(s_j - (w_{ij} + w_{ji}))}. \quad (6)$$

This definition can also be used in weighted signed networks ($w_{ij} \in [-1, 1]$), with a modified definition of s_i ($s_i = \sum_j |w_{ij}| + \sum_j |w_{ji}|$). In this case, $C_e^{\mathcal{W}, \mathcal{D}}(i)$ varies in $[-1, 1]$. It is positive when positive triangles formed around the focal node outweigh negative ones. It equals zero when no triangles formed with the focal node or positive triangles and negative triangles are balanced.

4 Experiments and Analysis

In this section, we evaluate the proposed directed closure coefficient in real-world networks. First, we compare it with the classic directed clustering coefficient. Then, we show how it can be applied in link prediction to improve the performance.

4.1 Directed Closure Coefficient in Real-World Networks

Datasets. We run experiments on 12 directed networks from different domains:

1. Six social networks.
 - (a) Two friendship networks. ADO-HEALTH [22]: a positively weighted friendship network created from a survey; DIGG-FRIENDS [23]: an online friendship network of news aggregator Digg.
 - (b) Three trust networks. BTC-ALPHA [24]: a weighted and signed trust network of users on Bitcoin Alpha; EPINIONS [25]: a weighted and signed trust network of online product rating site Epinions; WIKI-VOTE [26]: a network describing Wikipedia elections.
 - (c) One communication network. COLLEGEMSG [27]: a network comprised of messages between students.
2. Two citation networks. ARXIV-HEPPH [28]: a citation network from arXiv; US-PATENT [29]: a citation network of patents registered in the US.
3. Two online Q&A networks. ASKUBUNTU and STACKOVERFLOW [30]: two networks from Stack Exchange.
4. Two other networks. AMAZON [31]: a network describing co-purchased products on Amazon; GOOGLE [32]: a hyperlink network.

Table 1. Statistics of datasets, showing the number of nodes ($|V|$), the number of edges ($|E|$), the average degree (\bar{k}), the proportion of reciprocal edges (r), the average directed clustering coefficient ($\overline{C_c^D}$), and the average directed closure coefficient ($\overline{C_e^D}$) defined in this paper. Datasets having timestamps on edge creation are superscripted by (τ). Positively weighted networks are superscripted by (+), and networks having both positive and negative weights are superscripted by (\pm).

Network	$ V $	$ E $	\bar{k}	r	$\overline{C_c^D}$	$\overline{C_e^D}$
COLLEGE _{MSG} ^{τ}	1,899	20,296	10.69	0.636	0.087	0.017
ADO-HEALTH ⁺	2539	12,969	5.11	0.388	0.090	0.071
BTC-ALPHA ^{\pm, τ}	3783	24,186	6.39	0.832	0.046	0.006
WIKI-VOTE	7,115	104K	14.57	0.056	0.082	0.017
EPINIONS ^{\pm, τ}	132K	841K	6.38	0.308	0.085	0.010
DIGG-FRIENDS ^{τ}	280K	1,732K	6.19	0.212	0.075	0.008
ARXIV-HEPPH	34,546	422K	12.2	0.003	0.143	0.053
US-PATENT	3,775K	16,519K	4.38	0.000	0.038	0.019
ASKUBUNTU ^{τ}	79,155	199K	2.51	0.002	0.028	2e−4
STACKOVERFLOW ^{τ}	2,465K	16,266K	6.60	0.002	0.008	2e−4
AMAZON	403K	3,387K	8.40	0.557	0.364	0.234
GOOGLE	876K	5,105K	5.83	0.307	0.370	0.097

Table 1 lists some key statistics of these datasets. We see that in all 12 networks, the average directed closure coefficient is less than the average directed clustering coefficient. In these types of networks, we may say that compared to a triangle formation from centre-node based open triads, fewer triangles are formed from the end-node based open triads. In some networks (ADO-HEALTH and AMAZON), the difference between them is not very big; while in Q&A networks, the difference is more than 40 times.

From the scatter plots of the local directed closure coefficient and the local directed clustering coefficient (Fig. 3), we can see their relationship more clearly. First, the Pearson correlation is positive but weak (ranging from 0.134 to 0.759). Secondly, similar networks exhibit similar relationships between the two variables, as in two trust networks BTC-ALPHA and EPINIONS, in two citation networks ARXIV-HEPPH and US-PATENT or in two Q&A networks ASKUBUNTU and STACKOVERFLOW.

4.2 Link Prediction in Directed Networks

Many studies [33–37] have shown that future interactions among nodes can be extracted from the network topology information. The key idea is to compare the proximity or similarity between pairs of nodes, either from the neighbourhoods [34, 35], the local structures [36] or the whole network [37].

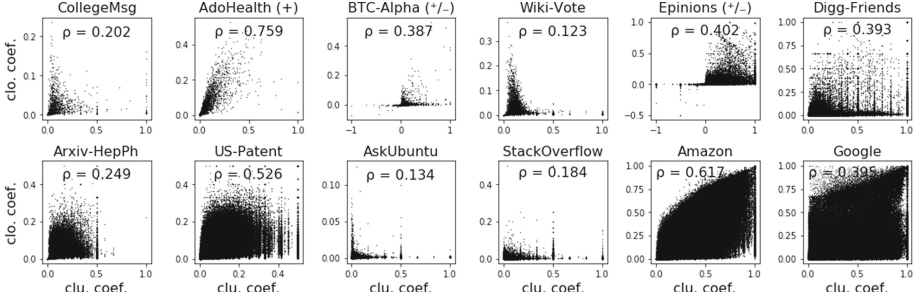


Fig. 3. Scatter plots of the local directed closure coefficient and the local directed clustering coefficient, with the Pearson correlation coefficient.

Baseline Methods. Most existing methods, however, focus solely on undirected networks. In this experiment, we show whether the information provided by the local directed closure coefficient can be used to enhance the performance of link prediction approaches for directed networks. As shown in [33], the neighbourhood based methods are simple yet powerful. We choose three classic similarity indices extended for directed networks as the baseline methods [38].

Let $N_{out}(i)$ be the out-neighbour set of node i (consisting of i 's successors); $N_{in}(i)$ be the in-neighbour set (consisting of i 's predecessors). The set of all neighbours $N(i)$ is the union of the two: $N(i) = N_{out}(i) \cup N_{in}(i)$. For an ordered pair of nodes (s, t) , the three baseline indices are defined: 1) Directed Common Neighbours index (DiCN): $DiCN(s, t) = |N_{out}(s) \cap N_{in}(t)|$, 2) Directed Adamic-Adar index (DiAA): $DiAA(s, t) = \sum_{u \in N_{out}(s) \cap N_{in}(t)} \frac{1}{\log |N(u)|}$, 3) Directed Resource Allocation index (DiRA): $DiRA(s, t) = \sum_{u \in N_{out}(s) \cap N_{in}(t)} \frac{1}{|N(u)|}$.

Proposed Indices. Combining the idea of the Common Neighbours index and the source and target closure coefficients (Definition 3), we propose two indices to measure the *directed closeness* in directed networks.

Definition 6. For an ordered pair of nodes (s, t) , the *closure closeness index*, denoted $CCI(s, t)$; and the *extra closure closeness index*, denoted $ECCI(s, t)$ are defined as:

$$CCI(s, t) = |N_{out}(s) \cap N_{in}(t)| \cdot (C_e^{src}(s) + C_e^{tgt}(t)),$$

$$ECCI(s, t) = |N(s) \cap N(t)| \cdot (C_e^{src}(s) + C_e^{tgt}(t)).$$

Different from the closure closeness index, the extra closure closeness index uses the set of all neighbours, because the source closure coefficient of node s and the target closure coefficient of node t can also bring in the direction inclination.

Setup. We model a directed network as a graph $G^D = (V, E)$. For networks having timestamps on edges, we order the edges according to their appearing times and select the first 50% edges and related nodes to form an ‘‘old graph’’, denoted $G_{old} = (V^*, E_{old})$. For networks not having timestamps, we randomly

choose 50% edges and related nodes as G_{old} and repeat 10 times in the experiment ($r_1 = 10$).

Let E_{new} be the set of future edges among the nodes in V^* , which is also what we aim to predict. Apparently, the total number of potential links on node set V^* is: $|V^*|^2 - E_{old}$. We apply each prediction method to output a list containing the similarity scores for all potential links in descending order, denoted L_p . An intersection of $L_p[0 : |E_{new}|]$ and E_{new} gives us the set of correctly predicted links, denoted E_{true} . The precision is then calculated by $|E_{true}|/|E_{new}|$.

For large networks ($|V| > 10K$), we perform a randomised breadth first search sampling of $5K$ nodes on G^D and repeat the above procedures r_2 times according to the size of the dataset. Therefore, for large networks without time-stamps we run the experiment $r_1 * r_2 = 10 * r_2$ times.

Table 2. Performance comparison of six methods on link prediction in directed networks (Precision %). RP (second column) gives the probability that a random prediction is correct. The best performance in each network is in bold type. The number at the foot of certain datasets indicates the total repeated times.

Network	RP	DiCN	DiAA	DiRA	CCI	ECCI
COLLEGEMSG ^τ	0.30	2.546	2.763	3.533	3.395	3.730
ADO-HEALTH ₍₁₀₎	0.10	8.404	8.406	8.304	10.23	11.07
BTC-ALPHA ^τ	0.05	8.588	9.269	7.313	8.418	9.226
WIKI-VOTE ₍₁₀₎	0.15	21.96	22.51	20.32	22.55	19.08
EPINIONS ₍₂₀₎ ^τ	0.37	3.613	3.662	3.531	3.490	5.106
DIGG-FRIENDS ₍₂₀₎ ^τ	0.33	6.649	6.709	6.685	7.135	5.569
ARXIV-HEPPH ₍₅₀₎	0.16	20.35	21.51	20.72	20.07	21.49
US-PATENT _(1,000)	0.04	9.787	10.14	9.987	11.67	11.31
ASKUBUNTU ₍₁₀₎ ^τ	0.03	4.100	4.912	4.163	5.412	4.697
STACKOVERFLOW ₍₁₀₀₎ ^τ	0.16	7.433	8.129	7.472	8.792	6.388
AMAZON ₍₅₀₀₎	0.06	23.71	27.94	27.43	26.76	29.46
GOOGLE ₍₅₀₀₎	1.19	44.48	52.32	50.29	49.39	46.24

Results and Discussion. We compare three baseline methods with two proposed methods (Definition 6) in Table 2. We see that the closure closeness index (CCI) has recorded the highest precision in 5 networks, and the extra closure closeness index (ECCI) has recorded the highest precision in 4 networks. It suggests that in most networks, including the local structure information of closure coefficient leads to improvement in link prediction. Sometimes the improvement is significant: In ADO-HEALTH and EPINIONS, ECCI is over 30% better than the baseline methods. In the other six networks (COLLEGEMSG, DIGG-FRIENDS, US-PATENT, ASKUBUNTU, STACKOVERFLOW and AMAZON), the precision of CCI or ECCI is over 5% higher than that of the baselines.

We also notice that in three networks (WIKI-VOTE, DIGG-FRIENDS and STACKOVERFLOW), where CCI records the highest precision, ECCI is, however, worse than the baseline methods. This suggests that sometimes the information provided by the extra neighbours without considering direction inclination conflicts with that provided by the source and target closure coefficients. Finding a method that better combines the information of common neighbours and closure coefficients is an interesting avenue for future study.

5 Conclusion

In this paper, we introduce the directed closure coefficient and its extension as another measure of edge clustering in complex directed networks. To better use it, we further propose the source and target closure coefficients. Through experiments on 12 real-world networks, we show that the proposed metric not only provides complementary information to the classic directed clustering coefficient but also helps to make some interesting discoveries in network analysis. Furthermore, we demonstrate that including closure coefficients in link prediction leads to significant improvement in most directed networks. We anticipate that the directed closure coefficient can be used as a descriptive feature as well as in other network analysis tasks.

Acknowledgement. This work was supported by the Australian Research Council, grant no. DP190101087: “Dynamics and Control of Complex Social Networks”.

References

1. Newman, M.E.: The structure and function of complex networks. *SIAM Rev.* **45**, 167–256 (2003)
2. Barabási, A.-L., et al.: *Network Science*. Cambridge University Press, Cambridge (2016)
3. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: simple building blocks of complex networks. *Science* **298**, 824–827 (2002)
4. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**, 814–818 (2005)
5. Solava, R.W., Michaels, R.P., Milenković, T.: Graphlet-based edge clustering reveals pathogen-interacting proteins. *Bioinformatics* **28**, i480–i486 (2012)
6. Noble, C.C., Cook, D.J.: Graph-based anomaly detection. In: *KDD* (2003)
7. LaFond, T., Neville, J., Gallagher, B.: Anomaly detection in networks with changing trends. In: *ODD² Workshop* (2014)
8. Henderson, K., Gallagher, B., Eliassi-Rad, T., Tong, H., Basu, S., Akoglu, L., Koutra, D., Faloutsos, C., Li, L.: RolX: structural role extraction & mining in large graphs. In: *KDD* (2012)
9. Musial, K., Juszczyszyn, K.: Motif-based analysis of social position influence on interconnection patterns in complex social network. In: *ACIHDS. IEEE* (2009)
10. Schall, D.: Link prediction in directed social networks. *Soc. Netw. Anal. Min.* **4**(1), 157 (2014)

11. Juszczyszyn, K., Kazienko, P., Musiał, K.: Local topology of social network based on motif analysis. In: KES. Springer (2008)
12. Rapoport, A.: Spread of information through a population with socio-structural bias: I. Assumption of transitivity. *Bull. Math. Biophys.* **15**, 523–533 (1953)
13. Wu, Z.-X., Holme, P.: Modeling scientific-citation patterns and other triangle-rich acyclic networks. *Phys. Rev. E* **80**, 037101 (2009)
14. Josang, A., Hayward, R.F., Pope, S.: Trust network analysis with subjective logic (2006)
15. Watts, D.J., Strogatz, S.H.: Collective dynamics of ‘small-world’ networks. *Nature* **393**, 440–442 (1998)
16. Onnela, J.-P., Saramäki, J., Kertész, J., Kaski, K.: Intensity and coherence of motifs in weighted complex networks. *Phys. Rev. E* **71**, 065103 (2005)
17. Zhang, B., Horvath, S.: A general framework for weighted gene co-expression network analysis. *Stat. Appl. Genet. Mol. Biol.* (2005)
18. Fagiolo, G.: Clustering in complex directed networks. *Phys. Rev. E* **76**, 026107 (2007)
19. Kunegis, J., Lommatzsch, A., Bauckhage, C.: The slashdot zoo: mining a social network with negative edges. In: WWW (2009)
20. Wang, J., Li, M., Wang, H., Pan, Y.: Identification of essential proteins based on edge clustering coefficient. *TCBB* **9**, 1070–1080 (2011)
21. Yin, H., Benson, A.R., Leskovec, J.: The local closure coefficient: a new perspective on network clustering. In: WSDM (2019)
22. Moody, J.: Peer influence groups: identifying dense clusters in large networks. *Soc. Netw.* **23**, 261–283 (2001)
23. Hogg, T., Lerman, K.: Social dynamics of Digg. *EPJ Data Sci.* **1**, 5 (2012)
24. Kumar, S., Hooi, B., Makhija, D., Kumar, M., Faloutsos, C., Subrahmanian, V.: REV2: fraudulent user prediction in rating platforms. In: WSDM. ACM (2018)
25. Massa, P., Avesani, P.: Controversial users demand local trust metrics: an experimental study on Epinions.com community. In: AAAI (2005)
26. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Predicting positive and negative links in online social networks. In: WWW (2010)
27. Panzarasa, P., Opsahl, T., Carley, K.M.: Patterns and dynamics of users’ behavior and interaction: network analysis of an online community. *J. Am. Soc. Inf. Sci. Technol.* **60**, 911–932 (2009)
28. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data (TKDD)* **1**, 2 (2007)
29. Hall, B.H., Adam, B.: 13 the NBER patent-citations data file: lessons, insights, and methodological tools. Patents, citations, and innovations: A window on the knowledge economy (2002)
30. Paranjape, A., Benson, A.R., Leskovec, J.: Motifs in temporal networks. In: WSDM (2017)
31. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. *ACM Trans. Web (TWEB)* **1**, 5 (2007)
32. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Statistical properties of community structure in large social and information networks. In: WWW (2008)
33. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.* **58**, 1019–1031 (2007)
34. Adamic, L.A., Adar, E.: Friends and neighbors on the web. *Soc. Netw.* **25**, 211–230 (2003)
35. Zhou, T., Lü, L., Zhang, Y.-C.: Predicting missing links via local information. *Eur. Phys. J. B* **71**, 623–630 (2009)

36. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: KDD (2014)
37. Meo, P.D.: Trust prediction via matrix factorisation. ACM Trans. Internet Technol. (TOIT) **19**, 1–20 (2019)
38. Zhang, X., Zhao, C., Wang, X., Yi, D.: Identifying missing and spurious interactions in directed networks. Int. J. Distrib. Sens. Netw. **11**, 507386 (2015)



Nondiagonal Mixture of Dirichlet Network Distributions for Analyzing a Stock Ownership Network

Wenning Zhang¹, Ryohei Hisano^{1,4}(✉), Takaaki Ohnishi^{2,4},
and Takayuki Mizuno^{3,4}

¹ Graduate School of Information Science and Technology,
The University of Tokyo, Tokyo, Japan

em072010@yahoo.co.jp

² Graduate School of Artificial Intelligence and Science,
Rikkyo University, Tokyo, Japan

³ National Institute of Informatics, Tokyo, Japan

⁴ The Canon Institute for Global Studies, Tokyo, Japan

Abstract. Block modeling is widely used in studies on complex networks. The cornerstone model is the stochastic block model (SBM), widely used over the past decades. However, the SBM is limited in analyzing complex networks as the model is, in essence, a random graph model that cannot reproduce the basic properties of many complex networks, such as sparsity and heavy-tailed degree distribution. In this paper, we provide an edge exchangeable block model that incorporates such basic features and simultaneously infers the latent block structure of a given complex network. Our model is a Bayesian nonparametric model that flexibly estimates the number of blocks and takes into account the possibility of unseen nodes. Using one synthetic dataset and one real-world stock ownership dataset, we show that our model outperforms state-of-the-art SBMs for held-out link prediction tasks.

Keywords: Block modeling · Edge exchangeability · Stock ownership

1 Introduction

Block modeling has been widely used in studies on complex networks [1, 2]. The goal of block modeling is to uncover the latent group memberships of nodes responsible for generating the complex network. The uncovered latent block structure is used for both prediction and interpretation. For prediction, block modeling is used to find missing or spurious edges [3, 4]. For interpretation, the estimated latent block structure provides a coarse-grained summary of the linkage structure that is particularly useful in complex networks, which is often messy at the primary level.

W. Zhang and R. Hisano—These authors made equal contributions.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021
R. M. Benito et al. (Eds.): COMPLEX NETWORKS 2020, SCI 943, pp. 75–86, 2021.
https://doi.org/10.1007/978-3-030-65347-7_7

The cornerstone model of block modeling is the stochastic block model (SBM) [5–7]. In the SBM, each node is assigned to a block. The edge probability in the network is governed solely by the linkage probability defined among these blocks. The goal of the SBM is to find the latent block structure and the linkage probability among the blocks. If given only one block structure, the model collapses to the Erdős–Rényi–Gilbert type random graph model that dates back to the 1950s [8,9].

The fact that the random graph model cannot reproduce basic properties, such as the sparsity and heavy-tailed degree distribution of complex networks, has always been an issue [1,10]. The failure of random graph models to reproduce these basic properties has recently been re-examined from the perspective of node exchangeability [11]. From the graphon formulation [12] and Aldous–Hoover theorem [13,14], it can be proven that the only possible network in the random graph model setting is either dense or empty [11,15]. This limitation makes the SBM rather unsuitable for modeling complex linkage patterns found in various complex networks.

Several authors have proposed models that go beyond such limitations using these modern findings. One line of research uses exchangeable point processes to generate the linkage patterns in a network [16]. In their formulation, edges appear when a pair of nodes occur in a nearby time position in the point processes. [16] showed that this formulation could generate sparse networks. Another line of research focuses on a more intuitive edge generation process based on edge exchangeability [11,15,17,18]. Edge exchangeable models have been proven to generate a sparse and heavy-tailed network. [19] proposed a model that considers the latent community structure in addition to the edge exchangeable framework. They called their model the mixture of Dirichlet network distributions (MDND) [19].

However, the MDND oversimplifies the latent block structure limiting it to only the diagonal case, similar to community detection algorithms. These limitations are problematic in instances in which the flow of influences (or information) among the blocks is the focus of research. One such example is the stock ownership network. In this setting, companies consider direct ownership and indirect ownership to maximize their influence and minimize risks [20]. A simple diagonal block structure only provides community-like clustering of companies, which is unsatisfactory.

In this paper, we provide a nondiagonal extension of the MDND (the NDMDND model) that makes it possible to estimate both the diagonal and nondiagonal latent block structure. Our model has no additional limitations than the MDND, and flexibly infers the number of blocks and considers the possibility of unseen nodes. It is noteworthy that our model can be regarded as a nonparametric extension of the sparse block model [21]. The sparse block model is a precursor model that focused on edge exchangeability even before the connection between sparse graphs and edge exchangeability was rigorously proven. We highlight both models in this paper.

2 Related Works

2.1 Sparse Block Model

In this section, we provide a brief explanation of the sparse block model. We use the notation (s_n, r_n) to denote the n th edge of the network, and $c_n = (c_{sn}, c_{rn})$ to denote the block pair to which that n th edge is assigned. We use A_k to define the node proportion distribution that captures which nodes are probable in block k . We use $Dir()$ to denote the Dirichlet distribution and $Cat()$ the categorical distribution, where the parameters are written inside the parentheses. We summarize the generative process as follows:

(A) Initialization

For each block pair $k = 1, \dots, K$,
we draw the node proportions $A_k \sim Dir(\tau)$

(B) Sampling of block pairs and edges

For each edge (s_n, r_n) ,
 (1) sample the block pair $c_n = (c_{sn}, c_{rn}) \sim Cat(\theta)$
 (2) sample the sender node from $s_n \sim Cat(A_{c_{sn}})$
 (3) sample the receiver node from $r_n \sim Cat(A_{c_{rn}})$.

Note that in the sparse block model, the latent block structure is defined in advance. The goal of the sparse block model is to infer the probability of each block to generate nodes (i.e., A_k), and the probability of each block pair (i.e., c_n) appearing from a given network. The fact that we have to specify the latent block structure is a huge limitation. It implies that we have to provide both the number of blocks to use and the block pairs' interaction patterns before seeing the data. Second, note that the same node pairs could appear multiple times in this setting (i.e., multigraph). These multiple edges could be used as a proxy for the edge weights. Although we could add a link function that links the proxy edge weights to the continuous edge weights, in this paper, we make the simple assumption that these multiple occurrences of an edge describe the weights of an edge. Finally, note also that the number of nodes used in the network is fixed; it does not increase as we sample more edges in the process.

2.2 Mixture of Dirichlet Network Distributions

The MDND is a nonparametric Bayesian model that attempts to infer the number of blocks from the observed network. Using a Bayesian formulation, it is also possible to estimate the probability of unseen nodes in sharp contrast to the sparse block model. The MDND assumes a diagonal block structure for the latent block structure and uses the Chinese restaurant process [22] to model the diagonal block pair linkage probability. The modeling of the probability of nodes given a block is more involved. Assume that a Chinese restaurant process for each block leads to each block's own set of nodes. For the model to force all the blocks to use the same set of nodes, we need to extend the Chinese restaurant

process to the Chinese restaurant franchise process [23]. The Chinese restaurant franchise process introduces an auxiliary assignment variable called a “table”. By separating the growth of the popularity of tables and the assignment of nodes (i.e., in [23]’s term “dish”) to the table, we can make multiple Chinese restaurant processes share the same set of nodes. We use $CRP(\alpha)$ to denote the Chinese restaurant process with hyperparameter α . We use subscripts to discern the multiple Chinese restaurant processes used in the model. We use s_{nt} and r_{nt} to denote the table assigned to the sending node that originates from the Chinese restaurant franchise process. α , τ , and γ are hyperparameters of the model. The generative process is as follows:

(A) Sampling of diagonal blocks

For each edge sample $c_n \sim CRP_B(\alpha)$ where c_{sn} is always equal to c_{rn}

(B) Sampling of edges

- (1) Sample a table for the sender node: $s_{nt} \sim CRP_{c_n}(\gamma)$
 if s_{nt} is a new table, then sample $s_n \sim CRP_N(\gamma)$
 else s_n is assigned the same node as s_{nt}
- (2) Sample a table for the receiver node: $r_{nt} \sim CRP_{c_n}(\gamma)$
 if r_{nt} is a new table, then sample $r_n \sim CRP_N(\gamma)$
 else r_n is assigned the same node as r_{nt} .

3 Nondiagonal Mixture of Dirichlet Network Distributions

3.1 Generating Process

Our proposed model, the NDMDND, can be considered as both a nonparametric Bayesian counterpart of the sparse block model and a nondiagonal extension of the MDND. Our model can be created by adding two components to the MDND: (1) adding another Chinese restaurant process that controls the number of block pairs used to model the latent block structure and (2) modifying the Chinese restaurant process that governs the appearance of blocks in the MDND to the Chinese restaurant franchise process. The latter extension is necessary because, as in the node-set case in the MDND, assuming a Chinese restaurant process separately for the sender blocks and receiver blocks would lead to each side having its own set of blocks. To prevent this, we need to make sure that both the sender and receiver sides share the same set of blocks. The node generation mechanism could be the same as in the MDND case without any further extension.

In the MDND, we need to add four Chinese restaurant processes: one for the block pair table (which we denote as $CRP_{block-pair}(\tau_{pair})$), one for the block tables for the sending nodes ($CRP_{block-send}(\tau_{block})$), one for the block tables for the receiving nodes ($CRP_{block-rece}(\tau_{block})$), and the last one responsible for generating the new blocks ($CRP_{block}(\gamma_{block})$). We use $c_{nt} = (c_{snt}, c_{rnt})$ to denote the pair table assigned to each edge. We further use s_{nbt} and r_{nbt} to denote the block tables assigned to the sender and receiver nodes, and s_{nb} and r_{nb} to denote the block assigned to each node. The generative process is as follows:

(A) Sampling of block pairs

For each edge sample pair table $c_{nt} \sim CRP_{block-pair}(\tau_{pair})$

if c_{nt} is a new pair table

(1) Sample $s_{nbt} \sim CRP_{block-send}(\tau_{block})$:

if s_{nbt} is a new send block table, then sample $s_{nb} \sim CRP_{block}(\gamma_{block})$

else assign the block associated to s_{nbt} to s_{nb}

(2) Sample $r_{nbt} \sim CRP_{block-rece}(\tau_{block})$:

if r_{nbt} is a new send block table sample $r_{nb} \sim CRP_{block}(\gamma_{block})$

else assign the block associated to r_{nbt} to r_{nb}

else assign the block table and block pair associated to the c_{nt} to

(s_{nbt}, r_{nbt}) and (s_{nb}, r_{nb})

(B) Sampling of edges

(1) Sample a table for the sender node: $s_{nt} \sim CRP_{c_n}(\gamma)$

if s_{nt} is a new table then sample $s_n \sim CRP_N(\gamma)$

else s_n is assigned the same node as s_{nt}

(2) Sample a table for the receiver node: $r_{nt} \sim CRP_{c_n}(\gamma)$

if r_{nt} is a new table, then sample $r_n \sim CRP_N(\gamma)$

else r_n is assigned the same node as r_{nt}

In NDMDND, γ_{block} controls the number of blocks used. A low γ_{block} with a relatively high τ_{pair} would lead to a more dense structure, whereas increasing γ_{block} would make the number of blocks increase. τ_{pair} and τ_{block} are trickier to interpret as both parameters also affect the possibility of considering a new block or block pair in the model. We further explain this issue in the next section.

3.2 Inference

The inference of NDMDND is rather involved compared with that of the MDND counterpart. In MDND, the direct sampling scheme is used to avoid the sampling of table assignments (Sect. 5.3 in [23]). However, in NDMDND, the sampling of both the table and table-to-block assignments turns out to be much simple (Sect. 5.1 in [23]). Moreover, a bonus of explicitly sampling tables is that we do not need to simulate the node counts (i.e., the number of tables with block k for a given node i , $\rho_{k,i}^{(1)}$ and $\rho_{k,i}^{(2)}$ in [19]) and instead evaluate them from our table assignments. We used these values to estimate the probability of a node appearing in an edge without block pairs. This probability is defined for both already seen nodes β_1, \dots, β_J and unseen nodes β_u . A simple sampling relation derives these β s: $\beta_1, \dots, \beta_J, \beta_u \sim \text{Dir}(\rho_{\cdot 1}^{(\cdot)}, \dots, \rho_{\cdot J}^{(\cdot)}, \gamma)$ where $\rho_{\cdot i} = \sum_k \rho_{k,i}^{(1)} + \rho_{k,i}^{(2)}$ represents the number of tables that a node $i (i \in \{1, \dots, J, J+1\})$ is selected in all the blocks.

Before introducing the inference scheme in more detail, we need to introduce some further notation. We use n_{t_p} , n_{t_s} , and n_{t_r} to count the number of edges or nodes assigned to a particular pair block table t , send block table s , and receive block table r , respectively. We use $n_{t_p}^{-i}$ to denote the count, ignoring the i th edge. We sometimes use the subscript i to denote the i th table, as in t_p^i , t_s^i , and $k_{t_s^i}$.

Algorithm 1. Inference algorithm of NDMDND

while not converged **do**
Update β s using \mathbf{t} and \mathbf{k} **for** $q = 1, \dots, T_1$ **do**Sample edge i at random

Sample from

$$p(t_p^i = t_p | t_p^{-i}, \mathbf{k}) \propto \begin{cases} n_{t_p}^{-i} f_{k_{t_s}, k_{t_r}}^{-s_i, -r_i}(s_i, r_i) \\ \tau_p p(s_i, r_i | t_p^{-i}, t_p^i = \text{new}, \mathbf{k}) \end{cases} \quad (1)$$

if $\hat{t}_p^i == \text{new}$ **then**

Sample from

$$p(t_s^i = t_s | t_s^{-i}, \mathbf{k}) \propto \begin{cases} n_{t_s}^{-i} f_{k_{t_s}}^{-s_i}(s_i) \\ \tau_p p(s_i | t_s^{-i}, t_s^i = \text{new}, \mathbf{k}) \end{cases} \quad (2)$$

if $\hat{t}_s^i == \text{new}$ **then**

Sample from

$$p(k_{t_s^i} = k | \mathbf{t}, k^{-t_s^i}) \propto \begin{cases} m_{k,k} f_k^{-s_i}(s_i) \\ \gamma_{\text{block}} f_{\text{new}}^{-s_i}(s_i) \end{cases} \quad (3)$$

if $\hat{k}_{t_s^i} == \text{new}$ **then**

Create a new block and assign the new block to the new table

elseAssign $\hat{k}_{t_s^i}$ to the new table**end if****else** $t_s^i = \hat{t}_s^i$ **end if**

... Perform exactly the same steps for the receiver blocks

elseAssign $t_p^i = \hat{t}_p^i$ and the accompanying send block table, and rece block table to t_s^i and t_r^i **end if****end for****for** $q = 1, \dots, T_2$ **do** Sample table number i from the sender tables

Sample from

$$p(k_{t_s^i} = k | \mathbf{t}, k^{-t_s^i}) \propto \begin{cases} m_k^{-t_s^i} f_k^{-s_{t_s^i}}(s_{t_s^i}) \\ \gamma_b f_{k_{\text{new}}}^{-s_{t_s^i}}(s_{t_s^i}) \end{cases} \quad (4)$$

... Perform exactly the same steps for the receiver block tables

end for**end while**

Furthermore, we use m_k to denote the number of tables associated with block k among both sender block tables and receiver block tables. Each send and receive block table is associated with a particular block. We use $k_{t_s^i}$ and $k_{t_r^i}$ to denote the block associated with the i th send block table and i th receive block table, respectively. We further use f to denote the likelihood and the symbol (i.e., “ $\hat{\cdot}$ ”) to represent the sampled value.

With this additional notation, we can outline the inference algorithm (Algorithm 1). In essence, the algorithm is a composition of the collapsed Gibbs sampling scheme. The first if-else branch considers whether to cluster the new edge to already existing block pair tables or create a new block pair table. If the latter is chosen, we have to consider two cases. One is to use existing block tables to generate the new block pairs, and the other is to create a new table to create the new block pair table. To judge whether to use existing block tables, we separate the sampling into sampling sender block tables and receiver block tables. If a new block table is chosen, we proceed in sampling a block assignment for the table (i.e., $k_{t_s^i}$ or $k_{t_r^i}$). The probability of assigning a new block is governed by γ_{block} . Algorithm 1 makes it clear that setting τ_{pair} too low would lead to the slow convergence of the MCMC. Therefore, in this paper, we set all the hyperparameters to $\tau_{pair} = 100$, $\tau_{block} = 10$, $\gamma_{block} = 10$, $\tau_{node} = 10$. The modification of the parameters did not change the main result in the paper provided τ_{pair} , τ_{block} , γ_{block} was sufficiently high for the sampler to find the correct block structure and sufficiently low for it not to outweigh the likelihood term.

4 Results

4.1 Dataset

Our experiments used two datasets: one containing synthetic data and the other containing real-world global stock ownership network data. The synthetic data was created, assuming the sparse block model. The stock ownership network is a subset of the Thomson Reuters ownership database. We focused on the ownership of significant assets in the second quarter of 2015. Both can be considered as a weighted network, and the datasets’ basic summary statistics are shown in Table 1¹. In both datasets, the network is sparse: the synthetic data has 7.2%, and the stock ownership data has 0.4% of all possible edges. Moreover, both datasets exhibit a heavy-tailed degree distribution, as shown in Fig. 1.

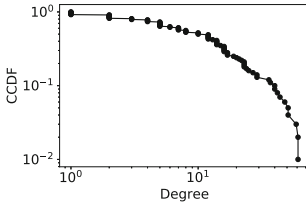
The motivation behind using a synthetic dataset is to illustrate whether our proposed algorithm recovers the ground truth block structure. In this experiment, we used all the edges in the synthetic data for training. Figure 2 shows the result of running the algorithm for 1,000 epochs². We confirm that after 100 epochs, the algorithm almost found the right block structure, and after 1,000 epochs, the result became more stable. Thus, we conclude that our model correctly uncovers the latent block structure of a given network.

¹ The weights for the stock ownership data is in percentage term.

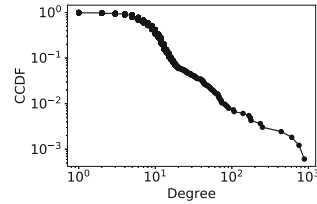
² One epoch comprises sampling all the edges in the training example once.

Table 1. Datasets

Dataset	Number of nodes	Number of edges	Min degree	Max degree	Min weight	Max weight
Synthetic	100	719	1	61	1	73
Ownership	1,639	10,465	1	886	1.0	69.7



(a) Synthetic data



(b) Stock ownership data

Fig. 1. Degree distribution

4.2 Quantitative Comparison

We compared the performance of NDMDND with that of five models: SBM [5, 24], degree corrected SBM [24, 25], weighted SBM [26], nested SBM [27], and MDND [19]. For SBM-type models, we used the state-of-the-art graph tool library [28], which uses the minimum description length principle to determine the number of blocks used in the SBM. Hence, it can be considered as a competitive alternative to the infinite relation model [29]. The degree corrected SBM further takes into account the heterogeneous degree distribution of nodes. For the weight function in the weighted SBM, we used the lognormal distribution for the synthetic data and an exponential distribution for the stock ownership data³. The nested SBM is a further extension of the SBM, which considers the fact that blocks themselves form a higher-level block structure. This additional layer may enhance the predictive probability of an unseen edge by taking into account the nodes that may be softly classified into multiple groups, akin to the mixed membership SBM [30].

We used a link prediction task as our basis for quantitative comparison. For both datasets, we used 80% of the data (i.e., edge list) as our training dataset and the remainder as our test dataset. We trained all our models using the training dataset and measured the model’s performance using the test dataset. The models that we compared have different likelihood functions. Some can even model edge weights. Hence, we compared the models using a simple binary classification task. For the stock ownership data, evaluating all the negative edges took so much time that it was impossible to assess the SBM models’ performance. Hence, we sampled 100,000 negative edges instead of using all the

³ We also tried the lognormal distribution for the stock ownership data, but it resulted in inferior performance.

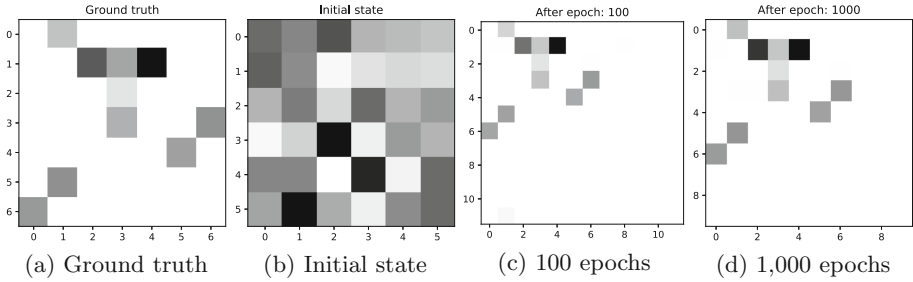


Fig. 2. Estimated block structure for the synthetic dataset

negative samples. A standard metric used in binary classification is the area under the receiver operating curve (AUC-ROC). However, the AUC-ROC overestimates the performance when the dataset is highly imbalanced, which applies to link prediction [31]. Moreover, theoretically, a model can only outperform in terms of the AUC-ROC when it outperforms in terms of the area under the precision-recall curve (AUC-PR) [32]. Therefore, we used the AUC-PR score for the primary comparison. Despite this, we also reported the AUC-ROC scores.

Table 2. Predictive performance

Model	Synthetic		Ownership	
	AUC-ROC	AUC-PR	AUC-ROC	AUC-PR
SBM	0.956	0.414	0.966	0.575
DCSBM	0.963	0.364	0.971	0.583
Nested SBM	0.969	0.412	0.974	0.599
Weighted DCSBM	0.971	0.672	0.97	0.568
MDND	0.918	0.298	0.893	0.477
NDMDND	0.983	0.736	0.968	0.673

Table 2 summarizes the results. It shows that NDMDND outperformed in terms of the AUC-PR quite significantly on both datasets. In terms of the AUC-ROC, all the models’ performance was almost the same, except for MDND, which was significantly inferior on both datasets. This inferior performance quite clearly highlights the limitations of the simple diagonal block structure, highlighting the importance of using our proposed NDMDND.

4.3 Estimated Block Structure

Figure 3 shows the estimated block structure for the stock ownership data. First, just by looking at the block structure matrix, we can see that several blocks are responsible for holding many of the other stocks in the dataset. The most prominent blocks are blocks 2 and 3, which hold many stocks in the dataset.

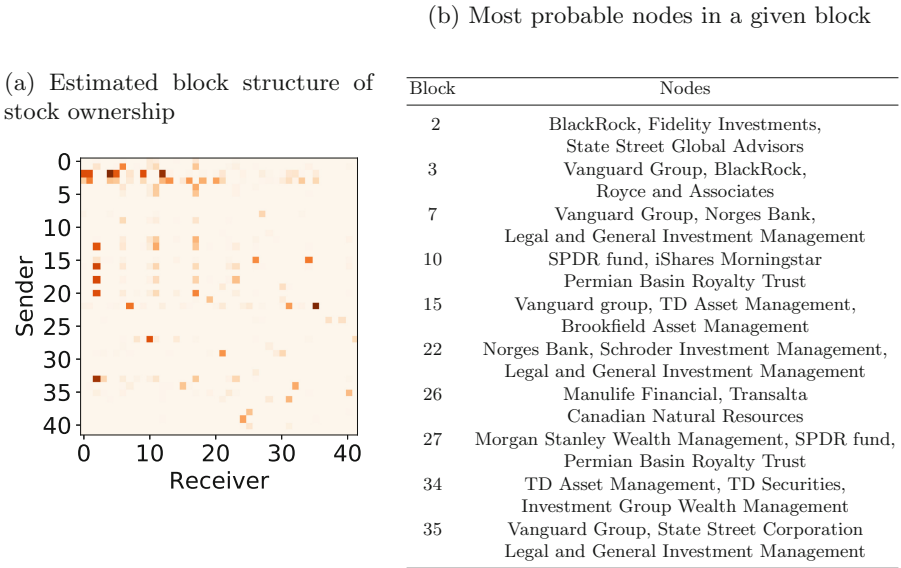


Fig. 3. Estimated results for the Reuters ownership dataset

As shown in Fig. 3, these two blocks include companies such as “BlackRock, Fidelity Investments, State Street Global Advisor,” which are famous global asset management companies. Another block pair that is quite huge in terms of the number of edges is block 22 to 35. Block 22 mainly contains European companies, whereas block 35 is a mixture of Canadian, U.S., and European asset managers. Another interesting block is block 26, which contains mostly Canadian companies owned by block 15. Block 15 is also mainly comprised of Canadian companies. Finally, block 10 does not own any stocks but is owned by many other nodes. This is not surprising because block 10 mainly comprises exchange-traded funds.

5 Conclusion

In this paper, we proposed an edge exchangeable block model that estimates the latent block structure of complex networks. Because the model is edge exchangeable, it reproduces the sparsity and heavy-tailed degree distribution that its random graph counterpart (i.e., SBM) fails to consider. We tested our model using one synthetic dataset and one real-world stock ownership dataset and showed that our model outperformed state-of-the-art models.

Acknowledgment. R.H. was supported by Grant-in-Aid for Young Scientists #20K20130, JSPS. T.O. was supported by Grant-in-Aid for Scientific Research (A) #19H01114, JSPS. We thank Maxine Garcia from Edanz Group for editing a draft of this manuscript.

References

1. Newman, M.E.J.: *Networks: An Introduction*. Oxford University Press, Oxford, New York (2010)
2. Doreian, P., Batagelj, V., Ferligoj, A.: *Advances in Network Clustering and Block-modeling*. Wiley, Hoboken (2019)
3. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.* **58**(7), 1019–1031 (2007)
4. Martínez, V., Berzal, F., Cubero, J.-C.: A survey of link prediction in complex networks. *ACM Comput. Surv.* **49**(4), 1–33 (2016)
5. Holland, P.W., Leinhardt, S.: An exponential family of probability distributions for directed graphs. *J. Am. Stat. Assoc.* **76**(373), 33–50 (1981)
6. Nowicki, K., Snijders, T.: Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *J. Classif.* **14** (1997)
7. Nowicki, K., Snijders, T.A.B.: Estimation and prediction for stochastic blockstructures. *J. Am. Stat. Assoc.* **96**(455), 1077–1087 (2001)
8. Erdős, P.: Graph theory and probability. *Can. J. Math.* **11**, 34–38 (1959)
9. Bollobás, B.: *Random Graphs*. Cambridge Studies in Advanced Mathematics, 2nd edn. Cambridge University Press, Cambridge (2001)
10. Barabási, A.-L., Albert, R.: Emergence of scaling in random networks. *Science* **286**(5439), 509–512 (1999)
11. Crane, H.: *Probabilistic Foundations of Statistical Network Analysis*. CRC Press, Boca Raton (2018)
12. Bickel, P.J., Chen, A.: A nonparametric view of network models and Newman–Girvan and other modularities. *Proc. Natl. Acad. Sci.* **106**(50), 21068–21073 (2009)
13. Aldous, D.J.: Representations for partially exchangeable arrays of random variables. *J. Multivariate Anal.* **11**(4), 581–598 (1981)
14. Relations on probability spaces and arrays of random variables. Institute for Advanced Studies
15. Crane, H., Dempsey, W.: Edge exchangeable models for interaction networks. *J. Am. Stat. Assoc.* **113**(523), 1311–1326 (2018)
16. Caron, F., Fox, E.B.: Sparse graphs using exchangeable random measures (2017)
17. Cai, D., Campbell, T., Broderick, T.: Edge-exchangeable graphs and sparsity. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 29, pp. 4249–4257. Curran Associates, Inc. (2016)
18. Crane, H., Dempsey, W.: Edge exchangeable models for interaction networks. *J. Am. Stat. Assoc.* **113**, 07 (2017)
19. Williamson, S.A.: Nonparametric network models for link prediction. *J. Mach. Learn. Res.* **17**(202), 1–21 (2016)
20. Garcia-Bernardo, J., Fichtner, J., Takes, F.W., Heemskerk, E.M.: Uncovering off-shore financial centers: conduits and sinks in the global corporate ownership network. *Sci. Rep.* **7**(1), 1–10 (2017)
21. Parkkinen, J., Gyenge, A., Sinkkonen, J., Kaski, S.: A block model suitable for sparse graphs (2009)
22. Phadia, E.: *Prior Processes and Their Applications*. Nonparametric Bayesian Estimation. Springer, Heidelberg (2013)
23. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical Dirichlet processes. *J. Am. Stat. Assoc.* **101**(476), 1566–1581 (2006)

24. Peixoto, T.P.: Nonparametric Bayesian inference of the microcanonical stochastic block model. *Phys. Rev. E* **95**, 012317 (2017)
25. Karrer, B., Newman, M.E.J.: Stochastic blockmodels and community structure in networks. *Phys. Rev. E* **83**, 016107 (2011)
26. Peixoto, T.P.: Nonparametric weighted stochastic block models. *Phys. Rev. E* **97**, 012306 (2018)
27. Peixoto, T.P.: Hierarchical block structures and high-resolution model selection in large networks. *CoRR*, abs/1310.4377 (2013)
28. Peixoto, T.P.: The graph-tool Python library. Figshare (2014)
29. Kemp, C., Tenenbaum, J.B., Griffiths, T.L., Yamada, T., Ueda, N.: Learning systems of concepts with an infinite relational model. In: *Proceedings of the American Association for Artificial Intelligence (AAAI)* (2006)
30. Airoldi, E.M., Blei, D.M., Fienberg, S.E., Xing, E.P.: Mixed membership stochastic blockmodels. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (eds.) *Advances in Neural Information Processing Systems*, vol. 21, pp. 33–40. Curran Associates, Inc. (2009)
31. Yang, Y., Lichtenwalter, R.N., Chawla, N.V.: Evaluating link prediction methods. *Knowl. Inf. Syst.* **45**(3), 751–782 (2014)
32. Davis, J., Goadrich, M.: The relationship between precision-recall and ROC curves. In: *Proceedings of the 23rd International Conference on Machine Learning, ICML 2006*, pp. 233–240. Association for Computing Machinery, New York (2006)



Spectral Clustering for Directed Networks

William R. Palmer^(✉) and Tian Zheng

Columbia University, New York, NY 10027, USA
wrp2110@columbia.edu

Abstract. Community detection is a central topic in network science, where the community structure observed in many real networks is sought through the principled clustering of nodes. Spectral methods give well-established approaches to the problem in the undirected setting; however, they generally do not account for edge directionality. We consider a directed spectral method that utilizes a graph Laplacian defined for non-symmetric adjacency matrices. We give the theoretical motivation behind this directed graph Laplacian, and demonstrate its connection to an objective function that reflects a notion of how communities of nodes in directed networks should behave. Applying the method to directed networks, we compare the results to an approach using a symmetrized version of the adjacency matrices. A simulation study with a directed stochastic block model shows that directed spectral clustering can succeed where the symmetrized approach fails. And we find interesting and informative differences between the two approaches in the application to Congressional cosponsorship data.

Keywords: Statistical network analysis · Community detection · Directed networks · Spectral clustering · Congressional cosponsorship

1 Introduction

The goal of community detection—one of the most popular topics in statistical network analysis—is to identify groups of nodes that are more similar to each other than to other nodes in the network. Determining the number of communities in a given network and the community assignments gives key insight into the network structure, creating a natural dimensionality reduction of the data. Moreover, the existence of clusters of highly connected nodes is a feature of many empirical networks ([6, 8]). Though there is growing research for directed networks ([10, 15]), community detection is best understood and most often implemented on undirected networks. In directed networks, edge directionality is often fundamental, and communities of nodes may be characterized by asymmetric relations. Consider, for example, citations, twitter follows and webpage hyperlinks. Properly accounting for edge directionality when analyzing such network data is very important.

Community detection is a clustering problem and requires an explicit notion of similarity between nodes. In general, clustering algorithms fall into two categories. There is model based clustering, which includes fitting procedures of

a model with well-defined clusters, and there are methods motivated by what clusters of the data objects should look like. These methods specify a related objective function, and partition the data to optimize it, often approximately. For points in \mathbb{R}^n , Gaussian mixture modeling falls in the first category, while k -means falls in the second. The most popular community detection algorithms, including spectral clustering [16] and modularity [6], fall in the second category. However, these methods have been shown to provide consistent clustering for certain random graph models ([2, 14]).

A broadly applicable method for clustering relational data, spectral clustering requires a similarity matrix between the data objects. For graph representations of network data, the adjacency matrix of edge weights provides measures of similarity between all nodes. Thus spectral clustering is a natural choice for community detection. Spectral clustering is particularly well understood in the symmetric, undirected setting [9]. The problem is more complicated in the more general setting of weighted, undirected networks, which we consider. Building from [21] and [3], this paper presents some of the theory of spectral clustering for directed networks, as well as two applications.

Section 1.1 motivates spectral clustering, Sect. 1.2 presents its general framework, and Sect. 2 explains our approach to spectral clustering for directed graphs. Sections 3 and 4 delve into applications—a stochastic block model simulation study and an analysis of recent cosponsorship data from the U.S. Senate.

1.1 Motivation

In order to motivate the use of spectral clustering for directed networks, we consider a toy example involving points in \mathbb{R}^2 .

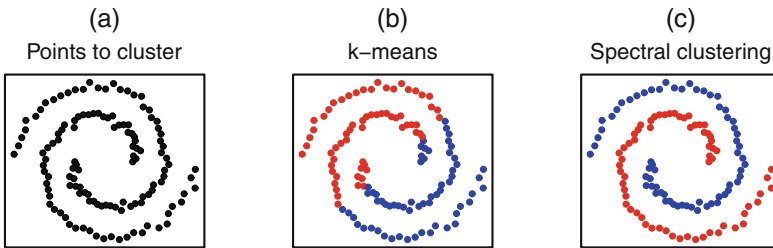


Fig. 1. Clustering of points in \mathbb{R}^2 with k -means and spectral clustering.

Figure 1a shows the points we wish separate into two spiral-shaped clusters. In Fig. 1b, k -means clustering fails to do this properly, since the clusters that we wish to capture have overlapping means. In Fig. 1c, spectral clustering properly separates the points.

Here we have defined a similarity matrix W_{ij} as the inverse Euclidean distance between points i and j if point j is among point i 's four nearest neighbors,

otherwise zero. The potential asymmetry of nearest neighbor relations means W is not symmetric, in general. Using our directed approach to spectral clustering, we are able to easily separate the points.

1.2 General Spectral Clustering Algorithm

Consider the problem of partitioning n individual entities into k subsets. Generally spectral clustering ([1,9]) proceeds in this way:

Algorithm 1. General spectral clustering

1. From data, construct an similarity matrix $W \in \mathbb{R}^{n \times n}$, where $W_{ij} \geq 0$ and $W_{ii} = 0$.
2. Compute a Laplacian $L \in \mathbb{R}^{n \times n}$ from W .
3. Compute first k eigenvectors of L , and combine into matrix X .
4. Cluster rows of X by k -means, or some other unsupervised algorithm.
5. Assign the original i^{th} entity to cluster ℓ iff the i^{th} row of X is assigned to ℓ .

Within this general framework, different approaches depend on the choice of Laplacian L , the inferring of k , manipulating of the eigenvectors in step 3 and the clustering method in step 4. Notable variations include principled eigenvector selection and clustering by Gaussian mixture modeling in [19], and Spectral Clustering On Ratios-of-Eigenvectors (SCORE) in [7], which relates to our approach, as detailed below.

For network-as-graph data, we begin with an adjacency matrix, and can skip directly to step 2. However, for directed networks, this W is not symmetric, and thus complicates the choice of Laplacian L . In the following section we motivate a graph Laplacian for directed, weighted networks, building towards it from an objective function corresponding to a notion of how communities should behave.

2 Spectral Clustering for Directed Graphs

We begin with a directed, weighted graph $G = (\mathcal{V}, \mathcal{E})$ with n vertices, represented by the adjacency matrix W . For a given k , $2 \leq k \ll n$, we seek a ‘best’ partition S_1, \dots, S_k of \mathcal{V} , one that maximizes within-cluster similarity while minimizing between-cluster similarity. We consider a notion of similarity related to the behavior of a random walk on the vertices \mathcal{V} .

To introduce this random walk, we begin with a few assumptions. We assume G is strongly connected, that is, for all $i, j \in \mathcal{V}$ there exists a directed path $i \rightarrow j$. Note that breaking up a network into its connected components is a natural first step in community detection. We also assume that G is aperiodic. We define a transition probability matrix P by $P_{ij} := W_{ij}/d_i^{\text{out}}$, where $d_i^{\text{out}} = \sum_j W_{ij}$ is the weighted out-degree of node i . Note $P = D^{-1}W$, where D is diagonal with $D_{ii} = d_i^{\text{out}}$. P is an irreducible aperiodic stochastic matrix, and thus has a unique stationary vector $\pi > 0$ satisfying $\pi^T P = \pi^T$, $\sum_i \pi_i = 1$. We define $\mathbf{\Pi}$ to be the diagonal matrix with $\mathbf{\Pi}_{ii} = \pi_i$, which we will use in the sequel.

With P and π it is natural to define a random walk $(N_t)_{t \in \mathbb{N}}$ on \mathcal{V} . In particular we can initialize the random walk according to π and then transition between nodes according to P . For a network with strong community structure, we expect this random walk to stay within the true communities more often than move between them. This leads to a notion of a ‘good’ community S —given that the random walk is on one of its nodes, the probability that next step jumps to a different community, i.e. $\mathbb{P}(N_{t+1} \notin S | N_t \in S)$, should be relatively low. The sum of these conditional probabilities across all communities in a given k -partition provides an objective function to minimize. In particular, we wish to find community assignments that solve:

$$\min_{S_1, \dots, S_k} \sum_{1 \leq \ell \leq k} \mathbb{P}(N_{t+1} \notin S_\ell | N_t \in S_\ell). \quad (1)$$

It is important to note that this objective function measuring the community assignments S_1, \dots, S_k takes fully into account the directionality of edges in G . This follows because the random walk N_t comes from the asymmetric transition matrix $P = D^{-1}W$. This objective is equivalent to the normalized cut criterion $\text{NCut}(S_1, \dots, S_k)$ for directed graphs in [21].

In (1) we have a discrete, non-convex optimization problem that is not readily solvable. Searching over all k -partitions is computationally intractable for even small networks. For example, there are over 580 million ways to divide 20 objects into 3 non-empty sets! Seeking an approximation solution, we proceed by rewriting the optimization problem in a form with a convex relaxation.

From G and a k -partition S_1, \dots, S_k of $[n]$, we define $\mathbf{g} = [g^1 \dots g^k] \in \mathbb{R}^{n \times k}$ by

$$g_i^\ell = \begin{cases} \frac{\sqrt{\pi_i}}{\sqrt{\sum_{j \in S_\ell} \pi_j}} & \text{if } i \in S_\ell \\ 0 & \text{otherwise.} \end{cases}$$

This matrix encodes the node assignments of S_1, \dots, S_k , has orthonormal columns, and can be shown (we do not go through the details here) to satisfy the equality

$$\text{Tr}(\mathbf{g}^T L \mathbf{g}) = \sum_{1 \leq \ell \leq k} \mathbb{P}(N_{t+1} \notin S_\ell | N_t \in S_\ell), \quad (2)$$

where $L = I - \frac{\mathbf{\Pi}^{1/2} P \mathbf{\Pi}^{-1/2} + \mathbf{\Pi}^{-1/2} P^T \mathbf{\Pi}^{1/2}}{2}$ is the graph Laplacian matrix for directed networks first proposed in [3]. Thus the optimization problem (1) is rewritten as the minimization of the left hand side of (2). While this formulation is no easier to solve exactly, it has a natural convex relaxation:

$$\min_{\substack{V \in \mathbb{R}^{n \times k} \\ V^T V = I}} \text{Tr}(V^T L V).$$

Here we are minimizing a Rayleigh quotient, so that a solution is the matrix X with columns given by normalized eigenvectors corresponding to the k smallest eigenvalues of L .

What remains is to determine a clustering from these eigenvectors of L , which constitute a loose approximation to the highly structured \mathbf{g} . Hence step 4 of the spectral clustering algorithm, for which we use k -means to create a partition. Note that 0 is the smallest eigenvalue of L , corresponding to eigenvector $\sqrt{\pi}$. Now the stationary vector π describes the limiting behavior of the random walk N_t on \mathcal{V} and relates to the degree distribution of G . It seems reasonable to question whether clustering should depend on the stationary distribution π , since this limiting behavior may be ancillary to existing community structure.

These considerations motivate clustering the rows of a transformed version of X , $X^* = \mathbf{\Pi}^{-1/2}X$. Here the i^{th} entry of each eigenvector is divided by $\sqrt{\pi_i}$. The first column of $\mathbf{\Pi}^{-1/2}X$ will be constant and equal to one, and therefore can be discarded. This ‘dividing out’ of the leading eigenvector agrees with the SCORE method for undirected methods. In [7], it is shown that the largely ancillary effects of degree heterogeneity in the Degree Corrected Stochastic Block Model are effectively removed by taking such entry-wise ratios.

In practice, when applied to various networks induced by the congressional co-sponsorship data discussed below, the values of the objective function (1) are consistently lower when clustering on the rows of $\mathbf{\Pi}^{-1/2}X$ as opposed to X , suggesting better resulting communities.

We now present in full our approach to spectral clustering for directed networks. We begin with a weighted, directed graph G defined by the adjacency matrix W , and a specified number of communities k . This is a modified version of Algorithm 1, above.

Algorithm 2. Spectral clustering for directed networks

1. From W , compute P , $\mathbf{\Pi}$ and $L = I - \frac{\mathbf{\Pi}^{1/2}P\mathbf{\Pi}^{-1/2} + \mathbf{\Pi}^{-1/2}P^T\mathbf{\Pi}^{1/2}}{2}$.
2. Compute the $k - 1$ eigenvectors corresponding to the 2nd- k^{th} smallest eigenvalues of L , and combine into matrix X .
3. Compute $X^* = \mathbf{\Pi}^{-1/2}X$, and normalize its columns.
4. Cluster rows of normalized X^* into k groups $1, \dots, k$ by k -means.
5. Assign i^{th} node of G to community ℓ if and only if i^{th} row of X^* is assigned to ℓ .

The computational complexity of this algorithm comes mostly from obtaining the k leading eigenvectors of L . The simple power method can be used to find leading eigenvectors, and when the adjacency matrix is sparse, as in many network applications, this complexity is slightly larger than $O(kn^2)$ ([7, 11]).

Note that when the adjacency matrix W is symmetric, we have that $L = I - D^{-1/2}WD^{1/2}$, which is precisely the normalized Laplacian L_{sym} for symmetric similarity matrices used in [1] and highlighted in [9]. This follows since $\pi^T = (d_1^{\text{out}}, \dots, d_n^{\text{out}}) / \sum_i d_i^{\text{out}}$ when $W^T = W$.

The question naturally arises as to how to choose k , the number of communities. This is an important question in all clustering problems. While there may exist prior knowledge about the true number of communities in a given network, often k is unknown, unfixed and needing to be learned from the data. In general, for clustering algorithms, there are many methods for choosing k . One

method devised particularly for spectral clustering is the eigengap heuristic [9]. It stipulates that we should choose k such that the first (smallest) k eigenvalues $\lambda_1, \dots, \lambda_k$ are relatively small, but λ_{k+1} is relatively large. We follow the eigengap heuristic in the applications below, choosing values of k such that $\lambda_{k+1} - \lambda_k$ is relatively large.

3 Simulation Study

We test the directed spectral clustering algorithm on networks simulated from a directed stochastic block model (SBM). Good performance on SBMs [18] is considered a necessary condition for useful community detection algorithms. However, block models do not account for complexities observed in many empirical networks, and thus do not alone provide sufficient criteria [14].

To generate a directed binary adjacency matrix $W \in \{0, 1\}^{n \times n}$, we assign n nodes randomly to communities 1, 2 and 3 with probabilities .3, .3 and .4, respectively, and then simulate an independent Bernoulli edge for each directed pair (i, j) , $i \neq j$ of nodes with probability $z_i^T Z z_j$, where

$$Q = \begin{bmatrix} .3 & .01 & .01 \\ .3 & .3 & .01 \\ .25 & .01 & .3 \end{bmatrix}$$

and z_1, \dots, z_n encode the community assignments.

We compare the performance of applying Algorithm 2 with W to an undirected approach in which we apply Algorithm 2 with $W_{\text{sym}} = W + W^T$. With this symmetrization, we effectively regard each directed edge as undirected. Using a naive graph transformation like W_{sym} is a common approach to community detection for directed networks ([10]). However, ignoring information about directionality can be problematic, and by using W_{sym} , we lose key information to help determine the correct k , and distinguish between communities 1 and 2.

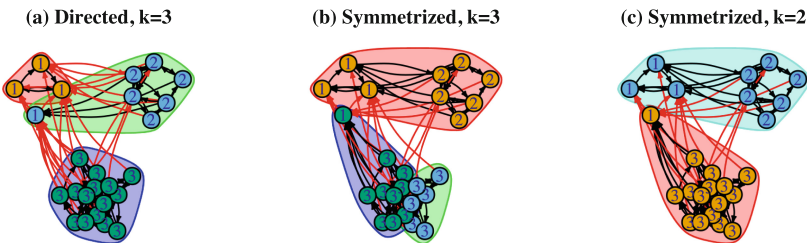


Fig. 2. Sociograms of the simulated network of size 25.

Figure 2 shows a single simulated network of size $n = 25$ along with the clustering results. Using the directed adjacency matrix and $k = 3$, Algorithm 2 nearly recovers the true communities, misclassifying just one node. On the other

hand, the results of Algorithm 2 with the symmetrized adjacency matrix W_{sym} and $k = 3$ combine nodes from communities 1 and 2, and split the nodes of group 3 into two clusters. For $k = 2$, the symmetrized approach nearly recovers group 3.

Increasing the size of the simulated network to $n = 100$ tells a somewhat similar story, with improvements for the symmetrized approach. Figure 3 shows the simulated network as adjacency matrix heatmaps. The block structure associated with the true groupings in Fig. 3a is clear. While node indices vary across the three panels, the estimated clusters in Fig. 3b–c are arranged to best align with the true blocks. Figure 3b shows again the near recovery of the true community structure by directed spectral clustering. In Fig. 3c it is clear that the symmetrized approach with continues to struggle to separate the communities correctly.

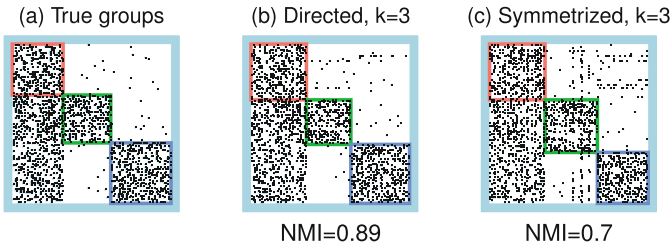


Fig. 3. Adjacency matrix heatmaps of a simulated network of size 100.

Across the bottom of Fig. 3b–c is the Normalized Mutual Information (NMI) measure between the true communities and the estimated clusterings. A value of 1 indicates exact agreement up to cluster relabeling. NMI is an information theoretic measure, relating the information needed to infer one cluster from the other. NMI satisfies desirable normalization and metric properties, and is adjusted for chance [17].

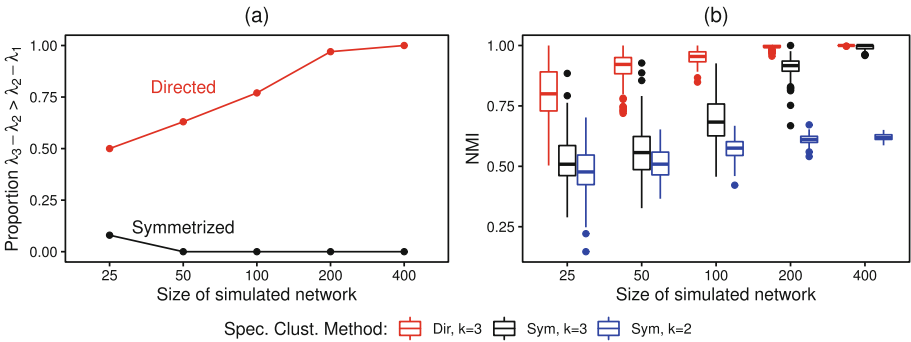


Fig. 4. Results of spectral approaches on 100 simulations at each network size. (a): Proportion of simulations where the eigengap heuristic correctly chooses $k = 3$ over $k = 2$. (b): NMI between true grouping and estimated clusterings.

Figure 4 summarizes results of the spectral approaches on 100 repeated simulations of the same stochastic blockmodel, for increasing numbers of nodes. Assuming the number of communities k is unknown, we would need to infer it from the data. Figure 4a shows the proportion of simulations where the eigengap heuristic correctly chooses $k = 3$ over $k = 2$. This is shown separately based on eigenvalues from the directed and symmetrized approaches. Under the directed approach, the rate at which the eigengap heuristic chooses correctly increases with the network size, reaching 100% for $n = 400$. On the other hand, under the symmetrized approach, the heuristic always chooses $k = 2$ over $k = 3$, for $n \geq 50$. Thus despite the success of the symmetrized approach as n increases (as seen in Fig. 4b), without prior knowledge, we would choose $k = 2$ communities rather than $k = 3$. Overall, Fig. 4b shows the superior performance the directed approach with $k = 3$, which consistently achieves an exact recovery of the true communities for $n \geq 200$.

We found that skipping step 3 of Algorithm 2, and not ‘dividing out’ the first eigenvector leads to better performance on these simulations. This makes sense because there is no degree heterogeneity within communities, and, moreover, community assignment is characterized by the in- and out-degree distributions. In such cases it is better to cluster the rows of X , not X^* .

4 Congressional Cosponsorship

Cosponsorship of bills in the U.S. Congress constitutes directed relational data. Previous network analysis of cosponsorship is found in [5]. Undirected modularity based community detection is applied to these networks in [20].

Every bill or amendment in Congress has one sponsor who introduces the measure, and may have one or more cosponsors, whose cosponsorship is generally viewed as an indication of support [12]. We represent cosponsorship of a bill as a set of directed binary edges from cosponsor to sponsor, one for each of the bill’s cosponsors, and we consider the weighted, directed graphs among members of Congress created by counting these directed binary edges across a set of bills and amendments. In this paper we analyze 21 months of Senate legislation from January 1, 2019 to September 30, 2020. This constitutes the data available at the time of writing from the 116th Congress. It includes 1,377 bills and amendments, from which we extract 7,667 cosponsorship edges.

The largest strongly connected component of the 116th Senate cosponsorship network includes 99 Senators, and contains 4,029 directed, weighted edges. We apply Algorithm 2 with the weighted directed adjacency matrix W , as well as with the naively symmetrized matrix $W_{\text{sym}} = W + W^T$. In both approaches, the eigengap heuristic does not provide strong evidence of community structure, with the first difference dominating. However, the second and third eigengaps are larger than the rest, indicating the possibility of $k = 2$ or $k = 3$ communities. Prior knowledge of the U.S. two party system along with current polarization points to $k = 2$; however, the persistent need for bipartisan legislation and the existence of moderate lawmakers on both sides suggests the possibility of $k > 2$.

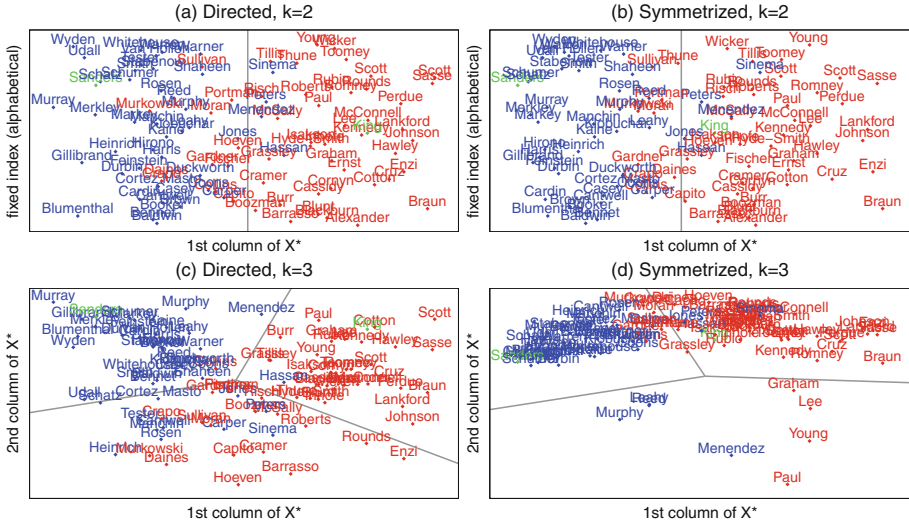


Fig. 5. Embeddings of Senators from ratios of eigenvectors of the Laplacian, with clustering boundaries from k -means.

Figure 5 shows results of the two spectral clustering approaches for $k = 2$ and $k = 3$ communities. Here we plot the columns of X^* from step 3 of Algorithm 2, along with the decision boundary separating the detected communities. The colors indicate party affiliation—blue for Democrat, green for Independent, and red for Republican. The results for $k = 2$ (Fig. 5a–b) are similar for the directed and symmetrized approaches, with, respectively, 85 and 86% of Senators clustered with the majority of their party, excluding independents.

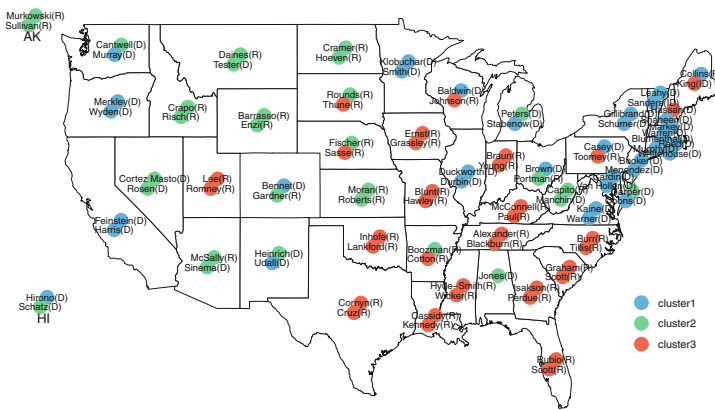


Fig. 6. Geographic relation of directed spectral clustering results for $k = 3$.

The results for $k = 3$ (Fig. 5c-d), however, differ greatly between the two approaches. The directed approach detects balanced and relatively well-separated communities, two of which align closely with party, and one that contains a mix of Republicans and Democrats mainly from the Plains, Mountain West, Southwest, and non-contiguous states. Figure 6 shows the full geographic correspondence of the detected communities. Meanwhile, the symmetrized approach detects one diffuse and separated community of four Democrats and four Republicans, and splits the remaining Senators roughly along the same lines as in Fig. 5b.

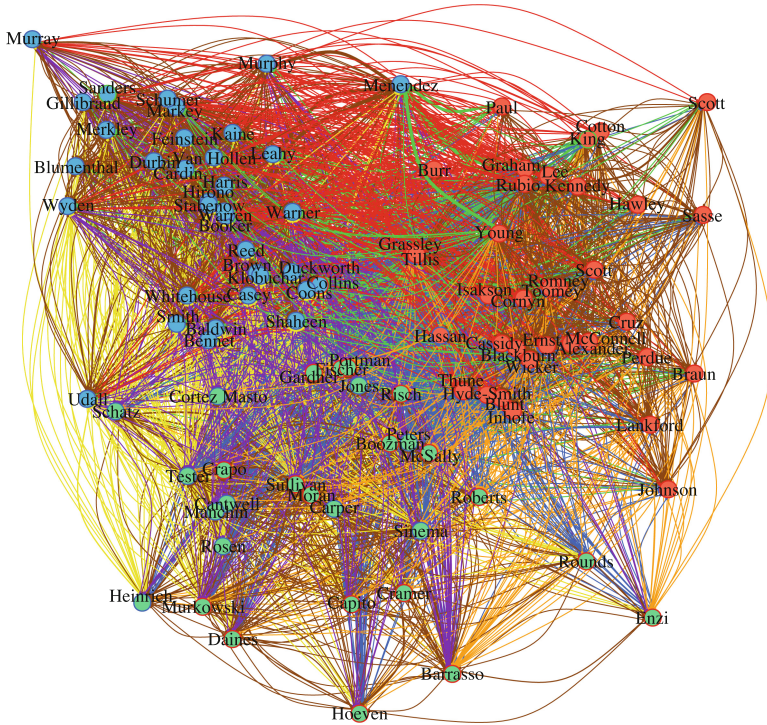


Fig. 7. Senate cosponsorship with clustering and embedding from Algorithm 2.

In Fig. 7 we use the same embedding as Fig. 5c to lay out a sociogram of the Senate cosponsorship network. In general, since spectral clustering methods provide embeddings, we can use them for visualization. The node interior coloring corresponds to detected communities, while the node outline color indicates party. Within cluster edges are brown, while between cluster edges are colored according to the community assignments of the cosponsor and sponsor nodes.

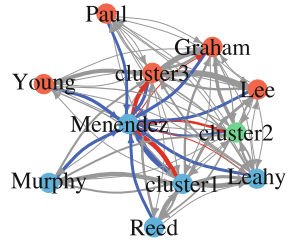
The imbalance of flows within and between clusters is apparent. We see a higher concentration of brown edges among the Republican core in the top right, and more inter-cluster out edges (purple and red) than inter-cluster in

edges (yellow and green) for the Democrats in the top left. These patterns are borne out more clearly by the inter and intra community intensities in Fig. 8a. Here we show the observed cosponsorship counts divided by the number of pairs of distinct legislators. The rows correspond, in order, to the blue (Democrat), green (mixed) and red (Republican) communities. The directed approach reflects inter-community asymmetries, while the symmetrized approach does not.

A notable feature of Fig. 7 and an exception to the patterns discussed above are four very prominent green edges from Republicans Graham, Lee, Paul and Young into Menendez (D-NJ) at the top middle, mirrored by three prominent brown edges into Menendez from Democrats Leahy, Murphy and Reed. These are precisely the eight Senators clustered together by the symmetrized approach with $k = 3$, appearing at the bottom of Fig. 5d. We isolate this star-like subnetwork in Fig. 8b. Here we include three nodes for the remaining Senators of each detected community and show the combined weighted edges between the eight individual Senators and these ‘remaining’ clusters. The edges flowing into Menendez are blue, those flowing out from Menendez are red, and the rest are grey.

1.3	0.87	0.66
0.54	1.17	0.77
0.38	0.42	0.92

(a) Inter and intra community intensities



(b) Collapsed subnetwork

Fig. 8. Further results of directed approach with $k = 3$.

Each blue edge from the Senators besides Menendez represents more than 23 cosponsorships, combining for a total of 174. Menendez is the minority ranking member of the Committee on Foreign Relations, and 169 of these cosponsorships involve international affairs legislation. Menendez cosponsors just 4 bills in return, and the ‘other seven’ have only 18 cosponsorships among themselves. Meanwhile, all four democrats exchange heavily with the remaining Senators in cluster 1, while the Republicans exchange with those remaining in cluster 3. Considering edge directionality, these eight Senators do not form a natural community within the context of the entire network. The directed approach reflects this, splitting these Senators along party lines. Unable to account for the patent asymmetry, the symmetrized approach allows the high weight of the edges flowing into Menendez to pull these Senators closer together, distorting the entire embedding, as seen in Fig. 5d, and classifying them as a separate community.

Data Note. Bill cosponsorship data is available from the ProPublica Congress API [13]. Amendment cosponsorship is obtained directly from Congress.gov [4].

5 Conclusion

In this paper we presented a variation of the general spectral clustering algorithm adopted for community detection on directed networks. We described the theoretical motivation behind the directed graph Laplacian, showing its connection to an objective function that reflects a notion of how communities of nodes in directed networks should behave. We applied our algorithm to simulated and empirical networks, and found encouraging and insightful results. When we ignore edge directionality by using a symmetrized adjacency matrix, we observe different results and worse performance on the simulated networks.

We see clear advantages to taking full account of the directionality of edges in complex networks. This is an important area of continued research, both from a theoretical and applied perspective.

References

1. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. *Neural Inf. Process. Syst.* **14**, 849–856 (2001). <https://doi.org/10.5555/2980539.2980649>
2. Bickel, P.J., Chen, A.: A nonparametric view of network models and Newman-Girvan and other modularities. *Proc. Nat. Acad. Sci.* **106**(50), 21068–21073 (2009). <https://doi.org/10.1073/pnas.0907096106>
3. Chung, F.: Laplacians and the Cheeger inequality for directed graphs. *Ann. Comb.* **9**(1), 1–19 (2005). <https://doi.org/10.1007/s00026-005-0237-z>
4. Congress.gov: Legislative search results. <https://congress.gov/search>
5. Fowler, J.H.: Connecting the congress: a study of cosponsorship networks. *Polit. Anal.* **14**(4), 456–487 (2006). <https://doi.org/10.1093/pan/mpj002>
6. Girvan, M., Newman, M.E.: Community structure in social and biological networks. *Proc. Nat. Acad. Sci.* **99**(12), 7823–7826 (2002). <https://doi.org/10.1073/pnas.122653799>
7. Jin, J.: Fast community detection by score. *Ann. Stat.* **43**(1), 57–89 (2015). <https://doi.org/10.1214/14-AOS1265>
8. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Statistical properties of community structure in large social and information networks. In: *Proceedings of WWW*, pp. 695–704 (2008). <https://doi.org/10.1145/1367497.1367591>
9. von Luxburg, U.: A tutorial on spectral clustering. *Stat. Comput.* **17**, 395–416 (2007). <https://doi.org/10.1007/s11222-007-9033-z>
10. Malliaros, F., Vazirgiannis, M.: Clustering and community detection in directed networks. *Phys. Rep.* **533**(4), 95–142 (2013). <https://doi.org/10.1016/j.physrep.2013.08.002>
11. Newman, M.E.: Modularity and community structure in networks. *Proc. Nat. Acad. Sci. U. S. A.* **103**(23), 8577–8582 (2006). <https://doi.org/10.1073/pnas.0601602103>
12. Oleszek, M.J.: Sponsorship and cosponsorship of house bills. *Congressional Research Service Report RS22477* (2019). <https://fas.org/sgp/crs/misc/RS22477.pdf>
13. ProPublica: Congress API (2020). <https://projects.propublica.org/api-docs/congress-api/>

14. Rohe, K., Chatterjee, S., Yu, B.: Spectral clustering and the high-dimensional stochastic blockmodel. *Ann. Stat.* **39**(4), 1878–1915 (2011). <https://doi.org/10.1214/11-AOS887>
15. Rohe, K., Chatterjee, S., Yu, B.: Co-clustering directed graphs to discover asymmetries and directional communities. *Proc. Nat. Acad. Sci. U. S. A.* **113**(45), 12679–12684 (2016). <https://doi.org/10.1073/pnas.1525793113>
16. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905 (2000). <https://doi.org/10.1109/34.868688>
17. Vinh, N.X., Epps, J.R., Bailey, J.: Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.* **11**, 2837–2854 (2010). <https://doi.org/10.5555/1756006.1953024>
18. Holland, P.W., Laskey, K.B., Leinhardt, S.: Stochastic blockmodels: first steps. *Soc. Netw.* **5**(2), 109–137 (1983). [https://doi.org/10.1016/0378-8733\(83\)90021-7](https://doi.org/10.1016/0378-8733(83)90021-7)
19. Xiang, T., Gong, S.: Spectral clustering with eigenvector selection. *Pattern Recogn.* **41**(3), 1012–1029 (2008). <https://doi.org/10.1016/j.patcog.2007.07.023>
20. Zhang, Y., Friend, A., Traud, A.L., Porter, M.A., Fowler, J.H., Mucha, P.J.: Community structure in congressional cosponsorship networks. *Physica* **387**(7), 1705–1712 (2008). <https://doi.org/10.1016/j.physa.2007.11.004>
21. Zhou, D., Huang, J., Schoelkopf, B.: Learning from labeled and unlabeled data on a directed graph. In: *Proceedings of the 22nd International Conference on Machine Learning* (2005). <https://doi.org/10.1145/1102351.1102482>



Composite Modularity and Parameter Tuning in the Weight-Based Fusion Model for Community Detection in Node-Attributed Social Networks

Petr Chunaev^(✉), Timofey Gradov, and Klavdiya Bochenina

National Centre for Cognitive Technologies, ITMO University,
Saint Petersburg, Russia

{chunaev,kbochenina}@itmo.ru, timogradov@yahoo.com

Abstract. The weight-based fusion model (WBFM) is one of the simplest and most efficient models for community detection (CD) in node-attributed social networks (ASNs) which contain both links between social actors (aka structure) and actors' features (aka attributes). Although WBFM is widely used, it has a logical gap as we show here. Namely, the gap stems from the discrepancy between the so-called Composite Modularity that is usually optimized within WBFM and the measures used for CD quality evaluation. The discrepancy may cause the misinterpretation of CD results and difficulties with the parameter tuning within WBFM. To fulfil the gap, we theoretically study how Composite Modularity is related to the CD quality measures. This study further yields a pioneering non-manual parameter tuning scheme that provides the equal impact of structure and attributes on the CD results. Experiments with synthetic and real-world ASNs show that our conclusions help to reasonably interpret the CD results and that our tuning scheme is very accurate.

Keywords: Community detection · Node-Attributed social network · Modularity · Parameter tuning · Weight-based fusion model

1 Introduction

Community detection (CD) in node-attributed social networks (ASNs) is an actively studied problem in social network analysis [4, 6] due to the necessity to explore a huge amount of real-world social network data containing both links between social actors (aka *network structure*) and actors' features (aka *network attributes*) such as age, interests, etc. While classical CD models deal either with the structure or the attributes, ASN CD methods aim at simultaneous usage or *fusion* of the both. The motivation behind it is that such a fusion may enrich the knowledge about ASN communities according to social science results [11].

A variety of different methods for ASN CD have been already proposed [4, 6]. Although they are widely used in applications, some of them have a serious logical gap that stems from the discrepancy between the objective functions within the CD optimization process and the functions used for CD quality evaluation [6]. Indeed, it is rather questionable that one function is optimized within the CD process and another function (that is not directly related to the latter) is used for evaluating the optimization results. We emphasize that we talk about structure- and attributes-aware quality measures and do not consider those estimating the agreement between the detected communities and the ground truth ones. There are specialized studies of the latter [6, Sect. 9] and we refer an interested reader to them. Note that the above-mentioned gap may cause misinterpretation of CD results and difficulties with the CD method parameter tuning.

In this paper, we reveal such a gap in the weight-based fusion model (WBFM) that is rather popular for ASN CD [6]. (We will give the description of WBFM under consideration in Sect. 2 and overview existing WBFMs in Sect. 3.) In particular, we show that there is a discrepancy between the so-called Composite Modularity that is usually optimized within WBFM and the corresponding CD quality measures (Modularity, Entropy, etc.), see Sect. 2. To fulfil the gap, we theoretically study how Composite Modularity is related to the CD quality measures called Structural and Attributive Modularities, where the latter is the substitution of Entropy (Sect. 4). From a more general point of view, we actually establish the connection between Modularities of two graphs and Modularity of the graph whose edge weights are linear combinations of edge weights of the two graphs. Our theoretical results further bring us to a simple parameter tuning scheme that provides the equal impact of structure and attributes on CD results (Sect. 5). It is worth mentioning that it is the first non-manual tuning scheme of this type within WBFM. Experiments with synthetic and real-world ASNs (Sect. 6) show that our conclusions allow for a reasonable interpretation of CD results within WBFM and that our tuning scheme is very accurate.

2 WBFM Within ASN CD Problem and Its Logical Gap

Below we first describe WBFM and the related CD problem. Then we recount the CD quality evaluation scheme within WBFM and reveal its logical gap.

2.1 Description of WBFM and Related ASN CD Problem

We model an ASN as *node-attributed graph* $G = (\mathcal{V}, \mathcal{E}, \mathcal{W}, \mathcal{A})$, where $\mathcal{V} = \{v_i\}_{i=1}^n$ is the set of nodes (social actors), $\mathcal{E} = \{e_{ij}\}$ the set of edges (links) between *all* nodes (i.e. $(\mathcal{V}, \mathcal{E})$ is complete), \mathcal{W} the set of edge weights¹, and \mathcal{A} the set of attribute vectors $A(v_i) = \{a_d(v_i)\}_{d=1}^D$, $v_i \in \mathcal{V}$, with *non-negative*² elements. Recall that $(\mathcal{V}, \mathcal{E})$ is called the *structure* and $(\mathcal{V}, \mathcal{A})$ the *attributes* of the ASN.

¹ An edge weight may be zero and this indicates that there is no social connection.

² If one deals with nominal or textual attributes, it is common to use one-hot encoding or embeddings to obtain their numerical representation.

The general WBFM may be thought to first convert G into the two weighted complete graphs by a certain rule: structural graph $G_S = (\mathcal{V}, \mathcal{E}, W_S)$ and attributive graph $G_A = (\mathcal{V}, \mathcal{E}, W_A)$, where $W_S = \{w_S(e_{ij})\}$ and $W_A = \{w_A(e_{ij})\}$ are the sets of edge weights in each graph. For convenience, we suppose that

$$\sum_{ij} w_S(e_{ij}) = 1, \quad \sum_{ij} w_A(e_{ij}) = 1. \quad (1)$$

Furthermore, the two graphs are fused to obtain the weighted graph $G_\alpha = (\mathcal{V}, \mathcal{E}, W_\alpha)$, where the elements of $W_\alpha = \{w_\alpha(e_{ij})\}$, with $e_{ij} \in \mathcal{E}$, are as follows:

$$w_\alpha(e_{ij}) = \alpha w_S(e_{ij}) + (1 - \alpha) w_A(e_{ij}), \quad \sum_{ij} w_\alpha(e_{ij}) = 1, \quad \alpha \in [0, 1]. \quad (2)$$

Here α is the *fusion parameter* that controls the impact of G_S and G_A . Note that $G_1 = G_S$ and $G_0 = G_A$ by construction.

Recall that *community detection* (CD) in G consists in unsupervised dividing \mathcal{V} into K disjoint³ *communities* $C_k \subset \mathcal{V}$, with $C = \{C_k\}_{k=1}^K$, such that $\mathcal{V} = \bigcup_{k=1}^K C_k$, and a certain balance between the following properties is achieved [4, 6]: (i) *structural closeness*, i.e. nodes in a community are more densely connected than those in different communities; (ii) *attributive homogeneity*, i.e. nodes in a community have similar attributes, while those in different ones do not.

As for WBFM, the CD problem consists in unsupervised dividing G_α into K disjoint communities $C_{k,\alpha} \subset \mathcal{V}$, with $C_\alpha = \{C_{k,\alpha}\}_{k=1}^K$, such that $\mathcal{V} = \bigcup_{k=1}^K C_{k,\alpha}$ and nodes in $C_{k,\alpha}$ are structurally close and homogeneous in terms of attributes.

Since one deals with a weighted graph G_α within WBFM, classical graph CD methods can be applied for finding C_α . A popular choice [6] is Louvain [3] aiming at maximizing *Modularity* [14], a measure of divisibility of a graph into clusters. In the context of WBFM, the maximization of Modularity of G_α is *implicitly* thought to provide structural closeness and attributive homogeneity also in G (similar schemes are applied e.g. in [7–10]).

2.2 WBFM CD Quality Evaluation Process and Its Logical Gap

Following the above-mentioned *implicit* thought, the partition C_α maximizing Modularity of G_α is treated as that for measuring structural closeness and attributive homogeneity in the initial G . Namely, C_α is used for calculating corresponding Modularity of G_S (a popular measure of structural closeness [4]) and Entropy of subsets of the corresponding attributes in \mathcal{A} (a popular measure of attributive homogeneity [4]). Thus one objective function is optimized to detect communities but the quality of the communities obtained is evaluated by measures not explicitly related to the objective function. This is the above-mentioned *logical gap* that may cause misinterpretation of CD results and difficulties with α -tuning in WBFM. To fulfil the gap, we will study how Modularity of G_α (provided by C_α) and C_α -based quality measures on G relate to each other.

³ Communities may be overlapping if necessary but here we focus on disjoint ones.

For completeness, let us first precisely define the involved measures (Modularity and Entropy). Note that the former works with graphs and the latter with sets of vectors. The definitions below are for G and C of a general form.

Modularity of graph G for a partition C is as follows:

$$Q(G, C) = \frac{1}{2m} \sum_{ij} (A_{ij} - \frac{1}{2m} k_i k_j) \delta_{ij} \in [-1, 1], \quad (3)$$

where i and j are from 1 to n ; A_{ij} is the edge weight between nodes v_i and v_j ; k_i and k_j are the weighted degrees of v_i and v_j , respectively; m is the sum of all edge weights in G ; and $\delta_{ij} = 1$ if c_i and c_j , the community labels of nodes v_i and v_j , coincide, and $\delta_{ij} = 0$ otherwise.

Modularity of graph G is then defined as $Q(G) = \max_C Q(G, C)$.

Entropy \mathcal{H} measures the degree of disorder of attribute vectors \mathcal{A} within communities. To unify notation, we define *Entropy of node-attributed graph G for a partition C* for the case of binary D -dimensional vectors as follows:

$$\mathcal{H}(G, C) = \sum_{C_k \in C} \frac{|C_k|}{|V|} H(C_k) \in [0, 1], \quad H(C_k) = - \sum_{d=1}^D \frac{\phi(p_{k,d})}{D \ln 2}, \quad (4)$$

where $\phi(x) = x \ln x + (1-x) \ln(1-x)$ and $p_{k,d}$ is the proportion of nodes in the community C_k with the same value on d th attribute.

Thus the CD task within WBFM is as follows: (i) one finds C_α maximizing Modularity of G_α , (ii) the C_α is used to calculate $Q(G_S, C_\alpha)$ and $\mathcal{H}(G, C_\alpha)$. Let us emphasize that this scheme is applied implicitly and there are no theoretical studies why this jump from $Q(G_\alpha, C_\alpha)$ to $Q(G_S, C_\alpha)$ and $\mathcal{H}(G, C_\alpha)$ is reasonable. We call this issue *Problem A* below. Another issue of the scheme (denoted by *Problem B* below) is that Entropy that deals with vectors is unnatural for WBFM aiming at representing G in a unified graph form. Moreover, Entropy may be not informative within WBFM as experiments in [5] and our own ones in Sect. 6.3 show. We will provide the solutions to *Problems A* and *B* in Sect. 4.

3 Related Works

WBFMs based on (2) have been widely tested on synthetic and real-world ASNs and have shown its superiority in CD quality to other CD models, see [1, 2, 7, 10, 12, 13, 15, 16]. Furthermore, there are many particular versions of (2) but it seems that the most balanced one is that from [5] with the following normalization:

$$w_S(e_{ij}) = \frac{\mu(e_{ij})}{\sum_{e_{ij} \in \mathcal{E}} \mu(e_{ij})}, \quad w_A(e_{ij}) = \frac{\nu(e_{ij})}{\sum_{e_{ij} \in \mathcal{E}} \nu(e_{ij})}, \quad (5)$$

where μ and ν are *structural* and *attributive* weight functions, correspondingly. Among other things, it is shown in [5] that (2) with (5) produces normalized versions of many existing WBFMs for different μ , ν and α .

The choice of α in (2) is difficult [4, 6]. In particular, there are no general parameter tuning schemes to make the impact of structure and attributes on

CD results equal. In fact, α is usually chosen manually and such a choice may be not fully justified. For example, WBFMs [1, 7, 13, 16] use $\alpha = 0$, while WBFMs [10, 12, 15] set $\alpha = 0.5$ in experiments, suggesting to achieve the equal impact.

What is more, we are unaware of any paper devoted to the study of the above-mentioned gap in WBFM, besides [6] where the problem is stated.

4 Theoretical Study

We first resolve *Problem B* by substituting Entropy (4) by $Q(G_A, C)$, i.e. Modularity of attributive graph G_A for a partition C . In these terms, if C is that maximizing $Q(G_A, C)$, then the links between nodes in each community in C have high attributive weights, i.e. the node attributes therein are homogeneous by construction. Additionally, the proposed measure works with graphs (oppositely to Entropy working with vectors) and naturally appears in WBFM as is seen from the results below. Moreover, it is more informative than Entropy as the experiments in Sect. 6.3 show.

Now we turn to *Problem A* about the connection of Modularity $Q(G_\alpha)$ and the CD quality measures (in our case, $Q(G_S, C_\alpha)$ and $Q(G_A, C_\alpha)$). The solution to *Problem A* is provided by the following theoretical results for a fixed G .

Theorem 1. *For any partition C , it holds that*

$$Q(G_\alpha, C) = \alpha Q(G_S, C) + (1 - \alpha)Q(G_A, C) + \alpha(1 - \alpha)Q(G_S, G_A, C), \quad (6)$$

where $Q(G_S, G_A, C)$ counts the difference of node degrees in G_S and G_A and is precisely defined in (9).

Proof. Fix a partition C . We first rewrite the ingredients of (3) in terms of (2):

$$\begin{aligned} A_{ij} &= \alpha w_S(e_{ij}) + (1 - \alpha)w_A(e_{ij}), & m &= \sum_{ij} w_\alpha(e_{ij}) = 1, \\ k_h &= \sum_{l, l \neq h} (\alpha w_S(e_{hl}) + (1 - \alpha)w_A(e_{hl})), & h &\in \{i, j\}. \end{aligned} \quad (7)$$

Furthermore, if $k_h^\star = \sum_{l, l \neq h} w_\star(e_{hl})$, where $h \in \{i, j\}$ and $\star \in \{S, A\}$, then

$$\begin{aligned} k_i k_j &= (\alpha k_i^S + (1 - \alpha)k_i^A) (\alpha k_j^S + (1 - \alpha)k_j^A) \\ &= \alpha^2 k_i^S k_j^S + \alpha(1 - \alpha) (k_i^S k_j^A + k_i^A k_j^S) + (1 - \alpha)^2 k_i^A k_j^A. \end{aligned} \quad (8)$$

If one takes (7) and (8) into account, (3) can be rewritten in the form

$$\begin{aligned} Q(G_\alpha, C) &= \alpha \cdot \frac{1}{2} \sum_{ij} (w_S(e_{ij}) - \frac{1}{2} \alpha k_i^S k_j^S) \delta_{ij} \\ &\quad + (1 - \alpha) \cdot \frac{1}{2} \sum_{ij} (w_A(e_{ij}) - \frac{1}{2} (1 - \alpha) k_i^A k_j^A) \delta_{ij} \\ &\quad - \alpha(1 - \alpha) \cdot \frac{1}{2} \sum_{ij} \frac{1}{2} (k_i^S k_j^A + k_i^A k_j^S) \delta_{ij}. \end{aligned}$$

Extracting $Q(G_S, C)$ and $Q(G_A, C)$ from this by (1) and (3) yields (6), where

$$Q(G_S, G_A, C) = \frac{1}{4} \sum_{ij} (k_i^S - k_i^A)(k_j^S - k_j^A) \delta_{ij}. \quad (9)$$

Theorem 2. For any partition C , it holds that

$$Q(G_S, G_A, C) = \frac{1}{4} \sum_{k=1}^K \left[\sum_{v_i \in C_k} (k_i^S - k_i^A) \right]^2 \geq 0, \quad (10)$$

$$Q(G_\alpha, C) \geq \alpha Q(G_S, C) + (1 - \alpha) Q(G_A, C). \quad (11)$$

This latter inequality is sharp for $\alpha = 0$ and $\alpha = 1$.

Proof. First note that

$$\sum_{i,j} k_i k_j \delta_{ij} = \sum_i k_i \sum_{j: c_j=c_i} k_j = \sum_{k=1}^K \left[\sum_{v_i \in C_k} k_i \sum_{v_j \in C_k} k_j \right] = \sum_{k=1}^K \left[\sum_{v_i \in C_k} k_i \right]^2.$$

What is more, $\sum_{i,j} k_i^S k_j^A \delta_{ij} = \sum_{i,j} k_i^A k_j^S \delta_{ij}$. Therefore by expanding (9) we get

$$\begin{aligned} Q(G_S, G_A, C) &= \frac{1}{4} \sum_{k=1}^K \left[\left[\sum_{v_i \in C_k} k_i^S \right]^2 - 2 \sum_{v_i \in C_k} k_i^S \sum_{v_i \in C_k} k_i^A + \left[\sum_{v_i \in C_k} k_i^A \right]^2 \right] \\ &= \frac{1}{4} \sum_{k=1}^K \left[\sum_{v_i \in C_k} (k_i^S - k_i^A) \right]^2. \end{aligned}$$

The last expression is non-negative. This fact and (9) yield (11). Finally, (11) follows from (6) by (11). The sharpness of (11) follows from (6).

Note that Theorems 1 and 2 connect Modularities of two graphs and Modularity of the graph whose weights are linear combinations of weights of the two graphs. It seems a key result for analysis of Modularity-based models for ASN CD.

We continue by introducing additional notation that simplifies further exposition. For the partition C_α such that $Q(G_\alpha) = Q(G_\alpha, C_\alpha)$, Theorem 1 gives:

$$\begin{aligned} Q_{\text{com}}^\alpha &= Q_{\text{str}}^\alpha + Q_{\text{attr}}^\alpha + Q_{\text{dif}}^\alpha, & Q_{\text{str}}^\alpha &= \alpha Q(G_S, C_\alpha) \\ Q_{\text{attr}}^\alpha &= (1 - \alpha) Q(G_A, C_\alpha), & Q_{\text{dif}}^\alpha &= \alpha(1 - \alpha) Q(G_S, G_A, C_\alpha), \end{aligned} \quad (12)$$

where we call $Q_{\text{com}}^\alpha = Q(G_\alpha)$ *Composite*, Q_{str}^α *Structural*, Q_{attr}^α *Attributive* and Q_{dif}^α *Differential* Modularity, correspondingly.

Thus within WBFM we maximize Composite Modularity Q_{com}^α that *consists not of the two components used for quality evaluation* (Structural Modularity Q_{str}^α and Attributive Modularity Q_{attr}^α) *but of additional* Differential Modularity Q_{dif}^α that counts the difference of node degrees in G_S and G_A . It moreover follows from Theorem 2 that WBFM under consideration does not provide optimal values of $Q_{\text{str}}^\alpha + Q_{\text{attr}}^\alpha$ for $\alpha \in (0, 1)$, if $Q_{\text{dif}}^\alpha \neq 0$. In particular, this means that WBFM is *at most equal in quality* to CD processes where $Q_{\text{str}}^\alpha + Q_{\text{attr}}^\alpha$ is optimized directly.

5 Parameter Tuning Scheme

We now propose a simple non-manual scheme for tuning α so that the impact of structure and attributes on CD results is equal. Since our terms are *unified* for

both the components, it is justified to define $\alpha = \alpha^*$ *providing the equal impact* as a solution to the equation

$$Q_{\text{str}}^\alpha = Q_{\text{attr}}^\alpha. \quad (13)$$

Theorem 3. *Let $Q_{\text{str}}^\alpha > 0$ and $Q_{\text{attr}}^\alpha > 0$ for any $\alpha \in [0, 1]$. If it holds that*

$$|Q(G_S, C_\alpha) - Q(G_S)| \leq \varepsilon Q(G_S), \quad |Q(G_A, C_\alpha) - Q(G_A)| \leq \varepsilon Q(G_A), \quad (14)$$

for some ε such that $0 \leq \varepsilon \ll 1$, then α^* satisfies the inequalities:

$$\frac{1 - \varepsilon}{1 + \varepsilon} \leq \alpha^* \cdot \frac{Q(G_S) + Q(G_A)}{Q(G_A)} \leq \frac{1 + \varepsilon}{1 - \varepsilon}. \quad (15)$$

Proof. We rewrite (13) by (12) as

$$\alpha = \frac{Q(G_A, C_\alpha)}{Q(G_S, C_\alpha) + Q(G_A, C_\alpha)}.$$

This and the conditions (14) immediately imply that (15) holds for α instead of α^* . Furthermore, the conditions (14) guarantee that Q_{str}^α and Q_{attr}^α are well-approximated uniformly for any $\alpha \in [0, 1]$ by $\alpha Q(G_A)$ and $(1 - \alpha)Q(G_S)$, correspondingly. These facts imply that (15) particularly hold for $\alpha = \alpha^*$.

As a consequence, Theorem 3 yields that for a small ε one can take

$$\tilde{\alpha} = \frac{Q(G_A)}{Q(G_S) + Q(G_A)} \quad (16)$$

as a good approximation for α^* providing the equal impact of structure and attributes on CD results. What is more, our experiments in Sect. 6.3 suggest that this is indeed so. It is interesting that (16) requires only the values of Modularities $Q(G_S)$ and $Q(G_A)$ to be applied.

Note that the proposed α -tuning scheme is the first *non-manual* one providing and giving *clear meaning* to the equal impact of the components within WBFM.

6 Experiments

Now we experimentally study the behaviour of the Modularities in (12). The source code and experimental results are presented on [Github](#). Below we use WBFM (2) with the normalization (5). The CD process in G_α is performed by Louvain [3] ([code](#)). Since different runs of Louvain may lead to different communities, we average the results over 5 runs and indicate the corresponding standard deviation. The fusion parameter α runs from 0 to 1 with step 0.05.

6.1 Synthetic Node-Attributed Networks

Recall that graphs G , G_S and G_A are complete and weighed according to the definitions in Sect. 2. Note that some edge weights may be zero and then one

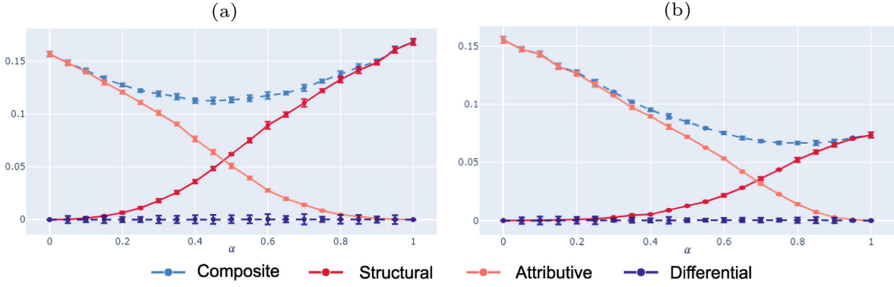


Fig. 1. Modularities in the graph pair ER+BA: (a) $Q(G_S) \approx Q(G_A)$ (ER-based G_S of 500 nodes and 6210 edges with equal non-zero weights, $p = 0.05$; BA-based G_A of 500 nodes and 6331 edges with equal non-zero weights, $m = 13$), (b) $Q(G_S) > Q(G_A)$ (ER-based G_S of 500 nodes and 24886 edges with equal non-zero weights, $p = 0.2$; BA-based G_A of 500 nodes and 6331 edges with equal non-zero weights, $m = 13$).

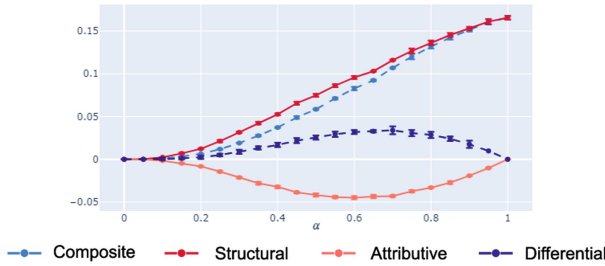


Fig. 2. Modularities for the experiment with the ER-based structural graph G_S (500 nodes and 6210 edges with equal non-zero weights, $p = 0.05$) and the star-based attributive graph G_A (500 nodes and 499 edges with equal non-zero weights).

should think that there is no structural or attributive link between the corresponding nodes. Below, if we say that a (complete) graph has M edges with non-zero weights, then edge weights in the graph are set zero for all edges except for M of them. Let us emphasize that the precise parameters of experiments necessary for reproducing the results are indicated in the figure captions.

Now we generate random graphs G_S and G_A by the well-known Erdős-Rényi (ER) and Barabási-Albert (BA) models which are standard for modelling social networks. The models are chosen as they produce graphs with different node degree distributions and thus may influence the behaviour of Differential Modularity Q_{dif}^α . What is more, we generate G_S and G_A in pairs ER+ER, ER+BA and BA+BA. In each pair we consider the following two cases: (a) when $Q(G_S)$ and $Q(G_A)$ are almost equal and (b) when one of $Q(G_S)$ and $Q(G_A)$ is greater than the other. These proportions can be achieved by varying the number of edges with equal non-zero weights in G_S and G_A .

It turns out that the results obtained in each pair are very similar qualitatively (and even quantitatively). For this reason we provide and analyze only

the results for the pair ER+BA, see Fig. 1. First note that in both cases Q_{dif}^α vanishes for all $\alpha \in [0, 1]$. Thus even the difference in degree distribution does not make its values large. It can be also observed that the intersection point of Structural and Attributive Modularities Q_{str}^α and Q_{attr}^α (corresponding to α^* that makes (13) valid) is closer to $\alpha = 1$, when $Q(G_S)$ is less than that of $Q(G_A)$, see Fig. 1(b). In the opposite case, it is close to $\alpha = 0$ (not shown), and is close to $\alpha = 0.5$, when the Modularities are almost equal, see Fig. 1(a).

The experiments performed so far hint that values of Q_{dif}^α are always vanishing. However, this is not true as Fig. 2 shows. In this experiment G_S is ER-based and G_A is a star graph, if one excludes the edges with zero weights. The difference in node degree distributions is so notable that the maximal value of Q_{dif}^α for $\alpha \in [0, 1]$ is rather separated from zero. This result emphasizes how interestingly WBFM may work for non-zero Q_{dif}^α . Note that Q_{str}^α and Q_{attr}^α have opposite signs for $\alpha \in [0, 1]$ here so that α^* providing the equal impact of structure and attributes may be thought to be 0.

6.2 Real-World Node-Attributed Networks

Below we use the undirected versions of the following publicly available datasets.

WebKB (Cornell, Texas, Washington, and Wisconsin) is a set of four networks, totally of 877 webpages with 1,608 hyperlinks gathered from universities websites. Each web page has a 1703-dimensional binary attribute vector whose each element indicates the presence of a certain word on that web page.

PolBlog is a network of 1,490 webblogs on US politics with 19,090 hyperlinks between these webblogs. Each node has a binary attribute describing its political leaning as either liberal or conservative.

Sinanet is a microblog user network extracted from weibo.com with 3,490 users and 30,282 relationships. Each node has a 10-dimensional positive numerical attribute vector describing user's interests.

Cora is a network of machine learning papers with 2,708 papers and 5,429 citations. Each node has a 1433-dimension binary attribute vector whose each element indicates the presence of a certain word in that paper.

In these experiments below, graphs G_S and G_A are constructed according to (2) and (5) with the following structural and attributive weight functions:

$$\mu(e_{ij}) = \begin{cases} 1, & \text{if } w(e_{ij}) = 1 \text{ in } (\mathcal{V}, \mathcal{E}), \\ 0, & \text{otherwise,} \end{cases} \quad \nu(e_{ij}) = \frac{A(v_i) \cdot A(v_j)}{\|A(v_i)\|_2 \|A(v_j)\|_2}.$$

Note that ν is the well-known Cosine Similarity and $\nu(e_{ij}) \in [0, 1]$ in our case as all the attributes are non-negative. It is worth mentioning also that the chosen μ and ν are among the most popular for this purpose [6] but, to be fair, the above-proved theorems stay valid for *any* non-negative weight functions.

The results for each of the four networks of WebKB are similar so we only present those for Washington, see Fig. 3(a). As in Sect. 6.1, one of $Q(G_S)$ and $Q(G_A)$ is greater than the other. However, $Q(G_S)$ is much greater than $Q(G_A)$ here so the case is almost degenerate. It can be observed that G_S has rather distinguishable communities, while G_A not. As a result, $\alpha^* \approx 0$ in this case.

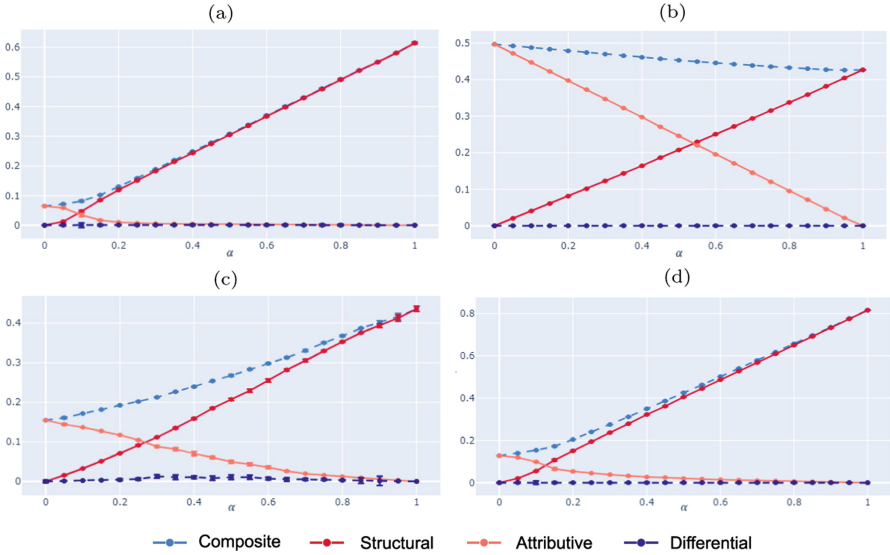


Fig. 3. Modularities for (a) WebKB Washington (b) PolBlogs (c) Sinanet (d) Cora.

Oppositely to WebKB, both G_S and G_A in PolBlogs have rather distinguishable communities, see Fig. 3(b). As for the attributes, it is indeed reasonable as the attributes in PolBlogs are one-dimensional and binary.

The results for Sinanet and Cora are correspondingly presented in Fig. 3(c) and (d) and are very similar qualitatively to those of WebKB. The values of $Q(G_S)$ are again greater than those of $Q(G_A)$ and this yields that $\alpha^* \approx 0$.

Surprisingly, we also note that $Q_{\text{diff}}^\alpha \approx 0$ for each chosen real-world network.

6.3 Evaluation of the Proposed Parameter Tuning Scheme and Attributes-Aware Modularity

To evaluate our parameter tuning scheme in Sect. 5, we calculate $\tilde{\alpha}$ by (16) and compare it with pre-calculated α^* , the solution to (13), for all the above-mentioned experiments, see Table 1. It turns out that absolute error does not exceed 0.01 in all the cases thus making the scheme very accurate for tuning α so that the impact of structure and attributes on CD results is equal.

Now we briefly compare the behaviour of Entropy $\mathcal{H}(G, C_\alpha)$ and Modularity $Q(G_A, C_\alpha)$ in the above-mentioned experiments with real-world networks (recall the definitions (3) and (4)). This is related to *Problem B* stated in Sect. 4. For clearer comparison, we introduce a slight variation of Entropy for a partition C , namely, $\tilde{\mathcal{H}}(G, C) = 1 - \mathcal{H}(G, C)$. This is more convenient as higher values of both $\tilde{\mathcal{H}}(G, C_\alpha)$ and $Q(G_A, C_\alpha)$ refer to higher attributive homogeneity.

We note that $\tilde{\mathcal{H}}(G, C_\alpha)$ and $Q(G_A, C_\alpha)$ have similar qualitative behaviour in all the experiments, namely, they decrease when α runs from 0 to 1, and it is fair for (2). However, their quantitative behaviour is rather different. Indeed, as

Table 1. Comparison of $\tilde{\alpha}$ in the proposed parameter tuning scheme and α^*

Experiment	Figure	$\tilde{\alpha}$	α^*	$ \tilde{\alpha} - \alpha^* $
ER+BR $Q(G_S) \approx Q(G_A)$	Fig. 1(a)	0.479	0.482	0.003
ER+BR $Q(G_S) > Q(G_A)$	Fig. 1(b)	0.688	0.678	0.010
ER+Star	Fig. 2	0.000	0.000	0.000
WebKB Washington	Fig. 3(a)	0.090	0.095	0.005
PolBlogs	Fig. 3(b)	0.540	0.540	0.000
Sinonet	Fig. 3(c)	0.262	0.268	0.006
Cora	Fig. 3(d)	0.126	0.135	0.009

Table 2. Entropy vs. Attributes-aware Modularity within WBFM

Network	$Q(G_A, C_0)$	$Q(G_A, C_1)$	$\tilde{\mathcal{H}}(G, C_0)$	$\tilde{\mathcal{H}}(G, C_1)$
WebKB Washington	0.065	0.002	0.9979	0.9898
PolBlogs	0.497	0.375	0.9999	0.9980
Sinonet	0.155	0.057	0.9996	0.9989
Cora	0.128	0.028	0.9999	0.9987

seen from Table 2, the rate of attributive homogeneity is hardly distinguishable in terms of $\tilde{\mathcal{H}}(G, C_\alpha)$ among the experiments (especially due to possible computational errors), while that in terms of $Q(G_A, C_\alpha)$ is explanatory. These facts provide new evidence that Entropy may be not informative within WBFM.

7 Conclusions

It is proved *analytically* in this paper that there is a logical gap in the well-known WBFM for ASN CD. This gap stems from the fact that optimal values of Composite Modularity optimized within WBFM do not generally provide those of Structural and Attributive Modularities that are the corresponding WBFM CD quality measures. Indeed, it turns out that Composite Modularity additionally includes *non-negative* Differential Modularity that may be very separated from zero in some special cases. At the same time, it is observed in experiments that it surprisingly vanishes in many cases of synthetic and real-world ASNs thus making WBFM optimal for providing the balance of structural closeness and attributive homogeneity in the above-mentioned terms.

Moreover, the identity for Composite Modularity and its usable terms proposed in this paper yield a simple and accurate parameter tuning scheme that gives *clear meaning* to and provides the equal impact of structure and attributes on the WBFM CD results. Note that it is the first non-manual one of this type.

Finally, it is also worth saying that we consider the theoretical results presented in this paper as a fundamental base for our current *analytical* comparative study of several Modularity-based ASN CD models in terms of CD quality. Such

a study can provide full generality of conclusions, oppositely to experimental comparative studies that are usually performed for ASN CD models, see e.g.. [6].

Acknowledgements. This research was financially supported by the Russian Science Foundation, Agreement 19-71-10078.

References

1. Akbas, E., Zhao, P.: Graph clustering based on attribute-aware graph embedding. In: Karampelas, P., Kawash, J., Özyer, T. (eds.) *From Security to Community Detection in Social Networking Platforms*, pp. 109–131. Springer, Cham (2019)
2. Alinezhad, E., Teimourpour, B., Sepehri, M.M., Kargari, M.: Community detection in attributed networks considering both structural and attribute similarities: two mathematical programming approaches. *Neural Comput. Appl.* **32**, 3203–3220 (2020)
3. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech. Theor. Exp.* **2008**(10), P10008 (2008)
4. Bothorel, C., Cruz, J., Magnani, M., Micenková, B.: Clustering attributed graphs: models, measures and methods. *Netw. Sci.* **3**(3), 408–444 (2015)
5. Chunaev, P., Nuzhdenko, I., Bochenina, K.: Community detection in attributed social networks: a unified weight-based model and its regimes. In: *2019 International Conference on Data Mining Workshops (ICDMW)*, pp. 455–464 (2019)
6. Chunaev, P.: Community detection in node-attributed social networks: a survey. *Comput. Sci. Rev.* **37**, 100286 (2020)
7. Combe, D., LARGERON, C., Egyed-Zsigmond, E., Gery, M.: Combining relations and text in scientific network clustering. In: *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2012*, pp. 1248–1253 (2012)
8. Cruz, J., Bothorel, C., Poulet, F.: Entropy based community detection in augmented social networks. In: *International Conference on Computational Aspects of Social Networks*, pp. 163–168 (2011)
9. Cruz, J., Bothorel, C., Poulet, F.: Détection et visualisation des communautés dans les réseaux sociaux. *Revue d'intelligence artificielle* **26**, 369–392 (2012)
10. Dang, T.A., Viennet, E.: Community detection based on structural and attribute similarities. In: *International Conference on Digital Society*, pp. 7–14 (2012)
11. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: homophily in social networks. *Ann. Rev. Sociol.* **27**(1), 415–444 (2001)
12. Meng, F., Rui, X., Wang, Z., Xing, Y., Cao, L.: Coupled node similarity learning for community detection in attributed networks. *Entropy* **20**(6), 471 (2018)
13. Neville, J., Adler, M., Jensen, D.: Clustering relational data using attribute and link information. In: *Proceedings of the Text Mining and Link Analysis Workshop, 18th International Joint Conference on Artificial Intelligence*, pp. 9–15 (2003)
14. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. *Phys. Rev. E* **69**, 026113 (2004)
15. Ruan, Y., Fuhry, D., Parthasarathy, S.: Efficient community detection in large networks using content and links. In: *Proceedings of the 22nd International Conference on World Wide Web, WWW 2013*, pp. 1089–1098 (2013)
16. Steinhäuser, K., Chawla, N.V.: Identifying and evaluating community structure in complex networks. *Pattern Recogn. Lett.* **31**(5), 413–421 (2010)



Maximal Labeled-Cliques for Structural-Functional Communities

Debajyoti Bera^(✉)

IIT -Delhi, New Delhi 110020, India
dbera@iiitd.ac.in

Abstract. Cliques are important building blocks for community structure in networks representing structural association between entities. Bicliques play a similar role for bipartite networks representing functional attributes (*aka.* labels) of entities. We recently proposed a combination of these structures known as labeled-cliques and designed an algorithm to identify them. In this work we show how to use these structures to identify structural-functional communities in networks. We also designed a few metrics to analyse those communities.

1 Introduction

A clique represents a set of mutually related entities in a network and has played an important role in *community detection* and *graph clustering* [6, 19]. Many network analysis methods, e.g., *clique-percolation method* [18] and *maximal clique centrality* [4], rely on the set of maximal cliques of a graph. Therefore, it natural to ask how to extend these results to networks with additional information.

One way to extend cliques would be to incorporate attributes on the nodes. The last decade has witnessed a massive increase in the collection of richer network datasets. These datasets not only contain the inter-entity relationships, but they also contain additional attributes (*aka.* “labels”) associated with each entity. For example, social network datasets contain both “structural relationships” (social links between users) and “functional attributes” (interests, likes, tags, etc.). A recent experimental study concluded that real-life communities are formed more on the basis of functional attributes of entities (like interests of users, functions of genes, etc.) rather than their “structural attributes” (those defined using cliques, cuts, etc.) [25]. Naturally, given *both* structural and functional information, we expect to find communities that are bonded on both.

The notion of cliques playing the role of seeds in a community structure ought to be strengthened if we also mandate functional similarity. In this work we address the question “*what is the role of such cliques in discovering cohesive structural-functional clusters?*” We are aware of only two prior solutions for this problem. Modani et al. [13] resolved the problem of finding “like-minded communities in a social network” by reducing it to that of finding maximal cliques in an unlabeled graph. Their solution was applying any graph clustering technique on a subgraph constructed using those maximal cliques. Motivated by a

similar problem, Wan et al. [24] studied the problem of finding communities that are strongly related in terms of both node attributes and inter-node relationships; their solution was a heuristic to avoid generating all maximal cliques. To the best of our knowledge, the first comprehensive graph-theoretic model for structural-functional clusters was given by Bera et al. [2] in the form of *maximal cliques of entities with a maximal set of shared labels*, aka. MLMCs. In that work the authors presented the idea, gave an algorithm to find those structures, and merely suggested a use for finding communities. In this work we outline tools and methods to employ MLMCs to analyse networks.

Overview of Results: We answer two specific questions. First, how to analyse a graph with the help of its MLMCs? In particular, what would be the statistics of MLMCs in a random graph? And, how far is a network from attaining stability, i.e., when the structural and functional linkages have converged to the same? To answer these questions, we propose a *null model* for labeled-graphs, and then use this null model to define *structural-functional divergence*.

The communities that we focus in this work are built on cliques; however, a clique in itself may be too strict a definition for a community. We devise an extension of the clique-percolation method [18] to labeled-graphs named **CBCPM** that incorporates similarity of labels also while constructing communities. For evaluating the functional cohesion of the communities found by our algorithm, we devise a new metric $\Phi\mathbf{C}$ to overcome a shortcoming of the *likemindedness* measure proposed earlier [13].

The interest in labeled graphs has recently gained popularity and there are now quite a few techniques for clustering them [1,5]. However, every clustering technique emphasises a different notion of community and it appears to be difficult to decide one clear winner. The relevance of this paper is limited only to the scenarios where clique-based communities are logical.

2 Background: Maximal-Labeled Cliques

We represent an undirected unweighted graph G by its sets of vertices and edges, i.e., $G = \langle V, E \rangle$. Similarly, we represent an undirected bipartite graph G by $G = \langle U, V, E \rangle$ where U and V represent the two sets of vertices and E represents the edges going between U and V . Suppose L is a finite discrete set of labels. A labeled-graph $G_L = \langle V, E, L, l \rangle$ is defined as a graph whose vertices have an associated subset of elements chosen from L . For any vertex v , $l(v) \subseteq L$ will be used to denote the labels of that vertex. A labeled-clique (LC) of G_L is defined to be any subset of vertices $V' \subseteq V$ and a subset of labels $L' \subseteq L$ such that (i) there is an edge between every pair of vertices in V' , and (ii) for every $v \in V'$, v is labeled using *all* the labels in L' ; we denote it $\langle L', V' \rangle$.

Our next notion is for unlabeled graphs that can be considered as a join of a bipartite graph and a general graph. Given a general graph $G_1 = \langle V, E_2 \rangle$ and a bipartite graph $G_2 = \langle U, V, E_1 \rangle$, a joined-graph is denoted by $\langle U, V, E_1, E_2 \rangle$ and defined as a network on U and V consisting of both sets of edges E_1 and E_2 . Observe that there are edges among vertices in V (E_2) and between vertices in

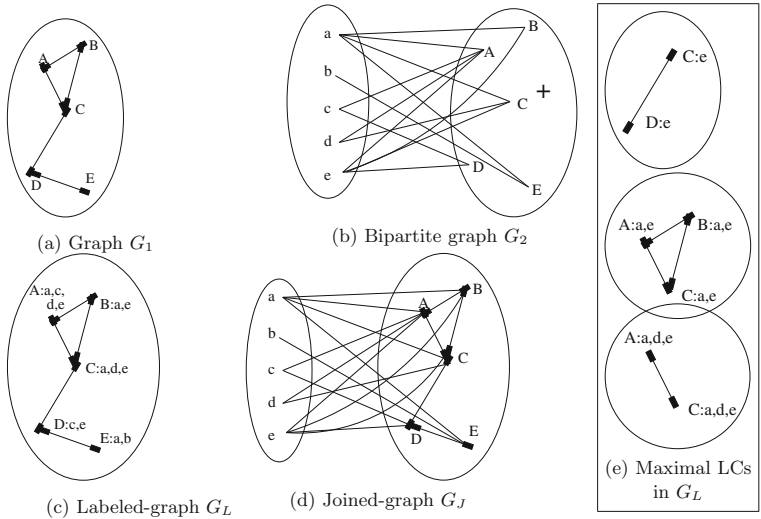


Fig. 1. Labeled-graph G_L combines G_1 and G_2 . G_J is the joined-graph representation of G_L . (Figure is reproduced from [2] with permission.)

U and V (E_1) but none among vertices in U . It was shown by Bera et al. [2] that a labeled graph can be treated as a joined graph and vice versa.

An MLMC—maximal clique with maximal set of labels, is a labeled-clique which does not remain an LC if we add any more vertex or label.

All of these concepts can be understood with the help of Fig. 1. It shows a network of entities $\{A, B, C, D, E\}$ as the general graph G_1 and Fig. 1b shows their association with labels from $\{a, b, c, d, e\}$ as the bipartite graph G_2 . Figure 1c shows a labeled-graph G_L that combines the information from G_1 and G_2 and $\langle\{a, e\}, \{A, C\}\rangle$ is an LC in G_L .

Examples: We present two examples to illustrate how MLMCs can help in analysing networks. Figure 2 presents the number-*vs*-size distribution of the MLMCs of two social-network datasets with tens of thousands of MLMCs and labelings (representing “user interests”) Not only the number of MLMCs of different sizes follow markedly different distributions, observe that the number of MLMCs with 5 (or 3 or 4) users are mostly same in the “Last.fm” dataset, whereas, the same number follows a rapidly decreasing trend in the “The Marker Cafe” dataset. Our explanation is that users of networks based on user-ratings (Last.fm) do not necessarily compare and correlate their ratings but users of a social network (The Marker Cafe) have a natural tendency to bond over shared interests. Such insights are attractive for targeted advertisement and personalized recommendation.

Table 1 shows some of the patterns we obtained by analysing the MLMCs of a DBLP dataset of papers published within 1984–2011 in data mining and related venues [23]—considering only the top venues and authors with 40+ papers in

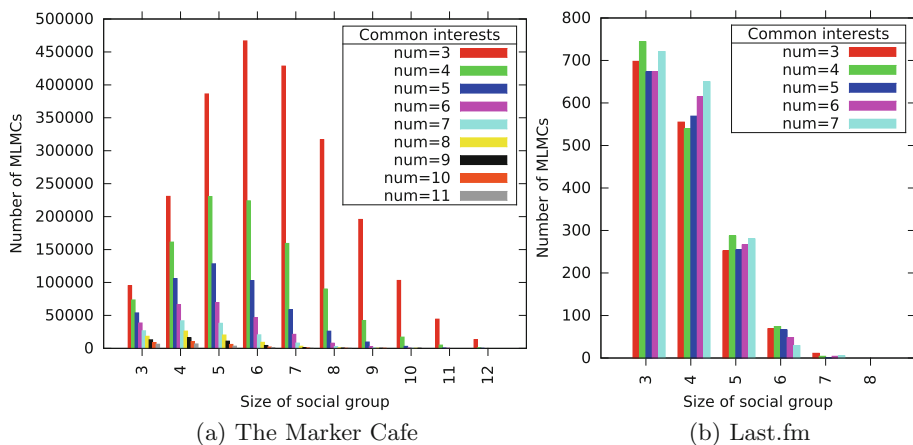


Fig. 2. MLMC profiles of social network datasets.

Table 1. Groups of prolific authors who share (pairwise) a common coauthor but are not collaborators despite having concurrent papers at common venues

Authors	@ Venues
Philip S. Yu, Heikki Mannila, Tao Li	TKDE(2008, 2009), Know. Inf. Sys. (2005– @ 2008, 2010, 2011), ICDM (2002, 2006), SDM(2008–2010)
Jian Pei, Christos Faloutsos, Wei Fan	ICDM(2005, 2006, 2008, 2010), @ SDM(2007, 2008, 2011), CIKM(2009), KDD(2004, 2006, 2008–2011)

them. We wanted to know which scientists are not collaborators but could easily be so. For that we constructed a labeled-graph of scientists in which the labels represented the venues of their papers. We linked two scientists if they have *do not* have a joint paper but share a common coauthor—roughly indicating a shared interest. We discovered 58 MLMCs that consisted of at least 3 authors and at least 10 venues; two such MLMCs are shown in Table 1. All such MLMCs represent potential collaborative groups that could have been formed due to familiarity (common coauthors) and concurrency (same venues).

3 Community Detection

Now we discuss how our labeled-cliques can help us find tightly bonded communities. Our objective is to establish *proof-of-concept application* of MLMCs; in reality, each network requires its own bespoke notion of community. The reader may refer to a recent survey [5] for many such techniques for labeled graphs.

3.1 Null Model for Labeled-Graphs

A common tool in network analysis is a *null model* that is a random graph with specific desirable properties. They are used to analyse networks, e.g., distinctness

of network from a randomly formed one, quality of a network clustering [7], etc. For example, the well-known notion of network modularity [16] uses a null model that preserves the expected degree of vertices. We use a null model that preserves the degree distribution of labeled-graphs. Given a labeled-graph $G = \langle V, E, L, l \rangle$, consider an equivalent joined-graph and denote its *bipartite component* as G^B and *general component* as G^N . We define null model for the labeled-graph G by simply joining the null models for G^N and G^B , which we describe below.

Null Model for G^N : For the non-bipartite component we use the well-studied *Configuration Model*(CM) [14,15] that creates a degree-preserving random graph. Consider a random approach that starts with an empty graph, picks two of the “unsaturated” vertices uniformly at random and connects them by an edge; a vertex is saturated when its number of edges equals its degree in G^N .

Null Model for G^B : We extend CM and generate a random graph with the same degrees as in G^B . The BiCM null model also generates graphs with the same properties [20], however, they use entropy-maximization unlike our combinatorial approach. We will, anyhow, denote our model too by BiCM.

We will follow the exact same approach as in CM and add edges between two randomly chosen unsaturated vertices, one each from L and V . Clearly, the final random graph has the same degrees as in G^B and also the same number of edges. Favoring simplicity, we allow the random graph to have multiple edges between vertices just like in CM.

Next we state a technical lemma on the expected number of common labels in BiCM. Consider any labeled-graph G from BiCM, and further, consider any two nodes $u, v \in V$ and any label $l \in L$. Let $N_{u,v}^l$ denote the indicator variable that is 1 iff l is the labeling of both u and v ; further, let $N_{u,v} = \sum_{l \in L} N_{u,v}^l$ denote the number of common labels. Let m denote the number of edges, d_u and d_v denote the degrees of u and v and c_l denote the number of nodes which have the label l .

Lemma 1. *The expected value of $N_{u,v}$ is*
$$\sum_{\substack{l \in L \\ c_l \geq 2}} \frac{1}{\binom{m}{c_l}} \sum_{\substack{r+g \leq c_l \\ r=1 \dots d_u \\ g=1 \dots d_v}} \binom{d_u}{r} \binom{d_v}{g}$$

$$\binom{m - d_u - d_v}{c_l - r - g}$$

Proof (Proof sketch). A standard approach is to attach $deg(x)$ stubs to a vertex x and connect to unassigned stubs at each step. Then the probability of selecting c_l stubs from the nodes, where there are d_u stubs from u , d_v stubs from v and $(m - d_u - d_v)$ other stubs, follows a trivariate hypergeometric distribution. $\mathbb{E}[N_{u,v}^l]$, which is same as the probability of selecting at least one stub of u and v each, can be now easily calculated from which the lemma follows.

An equivalent, but easier to compute, expression for $\mathbb{E}[N_{u,v}^l]$ can be obtained by applying the Chu-Vandermonde identity:

$$\mathbb{E}[N_{u,v}^l] = \left[\binom{m}{c_l} + \binom{m-d_u-d_v}{c_l} - \binom{m-d_u}{c_l} - \binom{m-d_v}{c_l} \right] / \binom{m}{c_l}$$

Algorithm 1. CBCPM: Finding overlapping SF clusters**Input:** Labeled-graph $G_L = \langle V, E, L, l \rangle$ **Output:** Overlapping clusters of V **Percolation parameters:** $k_l, k_s \in \mathbb{Z}^+$

- 1: $\mathcal{L} \leftarrow$ list of MLMCs of G_L with $\geq k_l$ labels & $\geq k_s$ vertices.
- 2: Form MLMC-overlap network \mathcal{N} :
- 3: Each node of \mathcal{N} is an MLMC M_i of \mathcal{L}
- 4: Edge between $M_i = \langle L_i, V_i \rangle$ & $M_j = \langle L_j, V_j \rangle$ if
- 5: $|L_i \cap L_j| \geq k_l - 1$ & $|V_i \cap V_j| \geq k_s - 1$
- 6: Obtain list \mathcal{C} of connected components of \mathcal{N}
- 7: **for all** connected component $C \in \mathcal{C}$ **do**
- 8: Output cluster $\{v : \exists \langle L', V' \rangle \in C, v \in V'\}$

3.2 Structural-Functional Divergence

The labeled-graphs represent two networks—one composed of structural links between nodes and another representing functional attributes. We conjecture that in many domains these two networks may converge with time as the nodes forge new structural links based on functional similarities or acquire new functionalities based on structural linkages. One way to measure the (dis)similarity of these two networks is to compare the general component with a monopartite projection of the bipartite component. For the latter, we fall back on the BiCM null model instead of other proposed approaches [11, 21]; the correct projection method really depends upon the application and was not investigated further. $\mathbb{E}[N_{u,v}]$ is computed on G^B in the definition below.

Definition 1. Given a labeled-graph $G = \langle V, E, L, l \rangle$, define its (λ, κ) -functional projection as an unlabeled graph G' on V in which an edge exists between u & v if $|l(u) \cap l(v)| \geq \min\{\lambda, \kappa \mathbb{E}[N_{u,v}]\}$. Let $CC(G)$ and $CC(G')$ denote the mean clustering coefficient of G and G' , respectively. (λ, κ) -structural-functional divergence of G is defined as: $\Delta_{\lambda, \kappa}^{SF}(G) = CC(G)/CC(G')$.

Choose some $\kappa > 1$. If there are $\kappa \mathbb{E}_{u,v}$ or more common labels between u and v , then this indicates a strong functional similarity between u and v when compared to the null model. The parameter λ is used for additional restrictions on the minimum functional similarity.

3.3 Structural-Functional Clustering

The *clique percolation method* (CPM) is a popular method for clustering of entities in a network considering only the structural links. This method identifies overlapping clusters which are composed of several (overlapping and maximal) cliques [18]. We are interested in clustering entities that are closely related both structurally and functionally. A previous approach by Modani et al. [13] first finds all MLMCs with a minimum number of nodes and common labels. Then it obtains the subgraph induced by the nodes of the MLMCs. They rightly claim that this subgraph is made up of those nodes that are better connected both

structurally and functionally. The authors then proposed to run any suitable overlapping (or non-overlapping) algorithm (e.g., CPM) on this subgraph.

However, we think better clusters can be obtained if the functional similarity is in-grained deeper in the cluster finding algorithm. Hence, we propose a ‘‘Cliques-Biclique Percolation Algorithm’’ (**CBCPM**) outlined in Algorithm 1. Like CPM, the clusters discovered by **CBCPM** are composed of maximal LCs. Each cluster is constructed from several LCs that are ‘‘connected’’—two LCs are said to be connected if they overlap in at least k_s nodes and at least k_l labels. The output of the algorithm are clusters of nodes from the connected components of the network of maximal LCs.

3.4 Quality of Structural-Functional Clustering

Finally, we study how to quantify the *quality of overlapping* clusters in a network. Following the approach of Modani et al., we consider one measure for the structural closeness of clusters and another for their functional similarity (or cohesion). If necessary, a weighted sum of both the measures can be used to construct a single measure of quality.

Suppose we are given clusters $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ in a labeled-graph $G_L = \langle V, E, L, l \rangle$ where C_i s are subsets of V , not necessarily disjoint. We will use e to denote the number of structural links in G_L . Let $\delta(u, v)$ denote an indicator variable for u and v co-occurring in some cluster together and similarly, $E(u, v)$ indicate an edge between u and v . $d(u)$ will denote the degree of a node $u \in V$ within the general component and $l(u)$ will denote its labels. For any label s , let $c(s)$ denote the set of users that have s as one of their labels. $comm(u_1, u_2, \dots)$ shall denote the set of clusters that contain all of the nodes u_1, u_2, \dots . Even though the clusters constitute only nodes, we will informally store the maximal set of common labels of all the nodes within each cluster.

Structural Quality: There are already a large number of options to choose from for structural quality. For our experiments, we chose a generalization of the highly popular Newman-Girvan ‘‘modularity’’ measure [16] that was proposed by Shen et al. [22]. These are built upon the notion of ‘‘coverage’’ and a null model. Coverage of a clustering is defined as the fraction of intra-cluster edges:

$$Cov(\mathcal{C}) = \frac{1}{2e} \sum_{C \in \mathcal{C}} \sum_{u, v \in C} E(u, v) = \frac{1}{2e} \sum_{u, v} E(u, v) \delta(u, v)$$

Modularity was initially defined for disjoint clusters. To apply this to overlapping clusters, a common trend is to use the notion of ‘‘belongingness’’ [17]. Shen et al. defined the contribution of a node u towards a cluster C as $\beta_{u,C} = \frac{1}{|comm(u)|}$ if $u \in C$ and 0 otherwise, and used it to define a generalized modularity OQ [22].

$$OCov(\mathcal{C}) = \frac{1}{2e} \sum_{C \in \mathcal{C}} \sum_{u, v \in C} E(u, v) \beta_{u,C} \beta_{v,C}$$

$$OQ(\mathcal{C}) = OCov(\mathcal{C}) - \mathbb{E}[OCov(\mathcal{C})] = \frac{1}{2e} \sum_{C \in \mathcal{C}} \sum_{u, v \in C} \left[E(u, v) - \frac{d_u d_v}{2e} \right] \beta_{u,C} \beta_{v,C}$$

Functional Quality: Despite several measures to quantify the similarity of nodes in a bipartite network, the only measure we found that was given explicitly for functional cohesion was “likemindedness” (LM) [13]. Let $\mathcal{S} : V \times V \rightarrow \mathbb{R}[0, 1]$ be a relevant measure for the functional similarity of two vertices, e.g., Jaccard similarity, Hadamard similarity, etc. Modani et al. defined likemindedness as the average similarity of all intra-cluster pairs of nodes (including pairs with duplicates, to remain consistent with modularity as hinted by the authors):

$$LM(\mathcal{C}) = \sum_{C \in \mathcal{C}} \sum_{u, v \in C} \mathcal{S}(u, v) / \sum_{u, v} \delta(u, v)$$

Consider a clustering in which there is one cluster with the two most similar nodes and all other nodes are in a single-member cluster each. It is easy to show that these clusters attain the maximum LM of $\max_{u \neq v} \mathcal{S}(u, v)$ among all clusterings. This led us to conclude that LM favors smaller, in fact, single or two membered, communities—not really a worthwhile measure of cluster quality.

This prompted us to define a new metric $\Phi\mathcal{C}$ for functional cohesion. First, we define “cohesion” of a clustering as the fraction of intra-cluster similarities over total similarity, enhanced with belongingness.

$$Coh^{\mathcal{S}}(\mathcal{C}) = \sum_{C \in \mathcal{C}} \sum_{u, v \in C} \mathcal{S}(u, v) \beta_{u, C} \beta_{v, C} / \sum_{u, v} \mathcal{S}(u, v)$$

Definition 2. For any similarity metric \mathcal{S} and a clustering \mathcal{C} of a labeled-graph, let $\mathbb{E}[Coh^{\mathcal{S}}(\mathcal{C})]$ be the expected cohesion in a corresponding BiCM random graph. Then functional modularity can be defined as: $\Phi\mathcal{C}^{\mathcal{S}}(\mathcal{C}) = Coh^{\mathcal{S}}(\mathcal{C}) - \mathbb{E}[Coh^{\mathcal{S}}(\mathcal{C})]$

Construct a complete weighted graph G' on V with weight of any edge (u, v) equal to $\mathcal{S}(u, v)$. By construction, the functional modularity on G is same as the overlapping modularity of G' .

For our experiments we used the Hamming similarity metric \mathcal{S}_H which is simply the fraction of labels that u and v have in common. Note that Coh and $\mathbb{E}[Coh]$ are not affected by the normalization factor. Instead, $\mathbb{E}[Coh]$ depends upon the edges which is governed by the null model. The following lemma will be useful in simplifying the denominator of $\mathbb{E}[Coh^{\mathcal{S}_H}]$. Recall that in the BiCM null model, the degree sequence of all nodes and all labels are fixed.

Lemma 2. Consider all graphs with a fixed set of labels, say L , and in which, $|c(l)|$ is fixed for every $l \in L$. Then,

$$\sum_{u, v} \mathcal{S}_H(u, v) = \frac{1}{\sigma} \sum_{l \in L: |c(l)| > 1} \binom{|c(l)|}{2}$$

The proof uses a simple double-counting of the nodes with a particular label. The denominator in $\mathbb{E}[Coh^{\mathcal{S}_H}]$ (and also in $Coh^{\mathcal{S}_H}$) therefore becomes a constant independent of the (random) graph. Furthermore, observe that $\mathbb{E}[\mathcal{S}_H(u, v)]$ in the random graph is same as $\mathbb{E}[N_{u, v}]$ in G^B (defined earlier).

Table 2. Labeled-graph datasets used for experimental evaluation.

Dataset	Type	Links represent ...	Labels represent ...	Nodes	Labels	Node links	Labelings	Num. of MLMC
Ning Creators' Net. (Ning) [12]	Social network	Friends	Group affiliation	11011	81	76262	4812	5459
'Café The-Markers's (CTM) [12]	Social network	Friends	Group affiliation	93664	88	1.74M	221610	34.7M
Ciao DVD (Ciao) [8]	Ratings of DVD reviews	Mutual trust	Reviews rated more than 2/5	20336	66109	7017	1.52M	79029
Filmtrust (FT) [9]	Movie ratings	Mutual trust	Movies rated more than 2/5	1530	1881	544	28580	1996
Last.fm (Lfm) [3]	Social net. of music listeners	Friends	Artists listened to	1892	17632	25434	92834	32344
Twitter-small (TwS) [10]	Social network	Mutual followers	Celebrities followed	1150	276	45360	42658	140M

Theorem 1. *Functional modularity of a clustering \mathcal{C} under Hamming similarity can be computed as:*

$$\Phi^{SH}(\mathcal{C}) = \sum_{C \in \mathcal{C}} \sum_{u, v \in C} [\mathcal{S}(u, v) - \mathbb{E}[N_{u, v}]] \beta_{u, C} \beta_{v, C} \Big/ \sum_{\substack{l \in L \\ |c(l)| > 1}} \binom{|c(l)|}{2}$$

4 Evaluation Results

To evaluate the effectiveness of our approaches, we applied them to several real-life datasets (described in Table 2). The “Twitter-small” dataset is constructed from the Twitter dataset [10] with edges representing “following a celebrity”; we selected as labels those users with followers between 15000 and 16000 (i.e., celebrities) and for nodes, those non-celebrities with 6000–65000 followers.

4.1 SF-Divergence

First we report the SF-divergence of our labeled-graph datasets in Table 3; we skip Ciao since it involved computing CC for a large number of nodes and labels which did not finish within a day.

A SF-divergence value less than one indicates that there are several nodes that share functionalities but are yet to form structural links. On the other hand, a value more than one indicates that nodes are yet to fully acquire functionalities from structurally connected nodes. We conjecture that the SF-divergence of a static social network (in which users are not joining or leaving) should approach

one in long term. We can see that the CTM and TwS networks display this behavior better than the other networks. This is expected for the TwS dataset since the “labels” in this network are celebrities and two users who follow each are more likely to follow the same celebrities. CTM users anyway show a highly “matured” behavior as was observed earlier in Fig. 2a.

4.2 Discovering Overlapping Communities

Now we report the quality of overlapping communities obtained by our CBCPM algorithm (Algorithm 1). Our goal was to show that, for similar setting of parameters k_s and k_l , CBCPM creates communities with better likemindedness than the existing CPMCore method [13] of running the CPM algorithm on the subgraph of nodes that are present in the MLMCs with at least k_l labels and k_s nodes. These parameters are related to the “percolation” of clique/labeled-cliques and has to be chosen carefully that was beyond our scope. Too large values may not find any community and too small values will create a single community. Therefore, we conducted experiments with different values of $k_s \geq 3$ and that of $k_l \geq 3$ and only considered clusters with at least two communities. We compared the overlapping modularity [22] (**OQ**) and the likemindedness [13] (**LM**) of the communities obtained by our CBCPM algorithm *vs.* those given by CPMCore [13]. We used the unnormalized Hamming similarity for $\mathcal{S}()$.

Table 3. SF-divergence values

Dataset	$\Delta_{2,3}^{SF}$
FT	0.28
Ning	0.39
Lfm	0.27
CTM	0.82
TwS	0.85

Table 4. Quality of Ning and FT communities

Dataset	Parameters	Method	OQ [22]	LM [13]
Ning (*)	$k_l = 3$ $k_s = 4$	CBCPM	0.05	3.38
		CPMCore	0.03	2.17
FT	$k_l = 3$ $k_s = 3$	CBCPM	0.35	5.27
		CPMCore	0.41	4.91
	$k_l = 4$ $k_s = 3$	CBCPM	0.27	5.51
		CPMCore	0.39	4.93

(*) Best k_s for $k_l = 3$ is used that maximized **LM**.

The Ning and the FT datasets generated very few MLMCs for some parameters. Therefore, we set $k_l = 3$, $k_s \geq 3$ for Ning which generated 118 MLMCs. Similarly, we used $k_l \geq 3$, $k_s = 3$ for the FT dataset that gave us 72 MLMCs. Results for the two clustering algorithms are presented in Table 4.

The quality measures of the larger Ciao and Lfm datasets are illustrated in Figs. 3 and 4, respectively. We tried several different values of k_l (indicated as CBCPM- k_l and CPMCore- k_l) and k_s (X-axis). We observed that CBCPM consistently found communities with higher LM compared to those found by CPMCore. Due to the stronger enforcement of functional similarity, CBCPM modularities are expected to be lower; however, we observed that the change is highly non-uniform here and sometimes even higher. We conclude that, in comparison to CPMCore, CBCPM finds communities with better functional qualities and with competitive structural qualities.

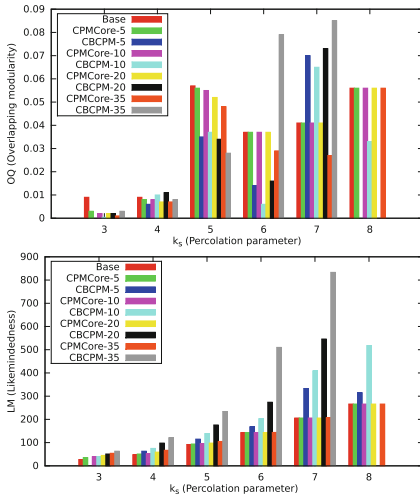


Fig. 3. Quality of CiaoDVD communities

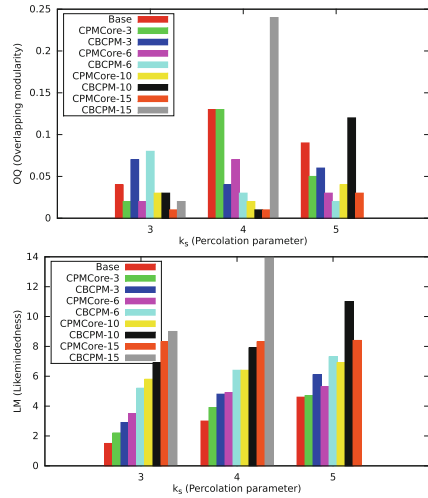


Fig. 4. Quality of Last.fm communities

5 Conclusion

Labeled-graphs are a richer representation of networks that can also store attributes of nodes, apart from the usual node-node relationship, and has been gaining popularity. In this work we show how to analyse the maximal labeled-cliques of these graphs, a concept that was recently introduced [2], and then show how to use those structures to identify clique-based communities. We also introduce a null model and a statistic to represent the attribute-level similarities within a community.

References

1. Baroni, A., Conte, A., Patrignani, M., Ruggieri, S.: Efficiently clustering very large attributed graphs. In: 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 369–376 (2017)
2. Bera, D., Esposito, F., Pendyala, M.: Maximal labelled-clique and click-biclique problems for networked community detection. In: 2018 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2018)
3. Cantador, I., Brusilovsky, P., Kuflik, T.: Second workshop on information heterogeneity and fusion in recommender systems. In: Proceedings of the 5th ACM Conference on Recommender Systems, (HetRec 2011) (2011)
4. Chin, C.H., Chen, S.H., Wu, H.H., Ho, C.W., Ko, M.T., Lin, C.Y.: cytohubba: identifying hub objects and sub-networks from complex interactome. BMC Syst. Biol. **8**(4), S11 (2014)
5. Chunaev, P.: Community detection in node-attributed social networks: a survey. Comput. Sci. Rev. **37**, 100286 (2020). <http://www.sciencedirect.com/science/article/pii/S1574013720303865>

6. Faghani, M.R., Nguyen, U.T.: A study of malware propagation via online social networking. In: *Mining Social Networks and Security Informatics*. Springer, Netherlands (2013)
7. Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**(3), 75–174 (2010)
8. Guo, G., Zhang, J., Thalmann, D., Yorke-Smith, N.: ETAF: an extended trust antecedents framework for trust prediction. In: *Proceedings of the 2014 International Conference on Advances in Social Networks Analysis and Mining* (2014)
9. Guo, G., Zhang, J., Yorke-Smith, N.: A novel Bayesian similarity measure for recommender systems. In: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence* (2013)
10. Kwak, H., Lee, C., Park, H., Moon, S.: What is Twitter, a social network or a news media? In: *Proceedings of 19th International Conference on World Wide Web* (2010)
11. Latapy, M., Magnien, C., Vecchio, N.D.: Basic notions for the analysis of large two-mode networks. *Soc. Netw.* **30**(1), 31–48 (2008)
12. Lesser, O., Tenenboim-Chekina, L., Rokach, L., Elovici, Y.: Intruder or welcome friend: inferring group membership in online social networks. In: *Social Computing, Behavioral-Cultural Modeling and Prediction* (2013)
13. Modani, N., et al.: Like-minded communities: bringing the familiarity and similarity together. *World Wide Web* **17**(5), 899–919 (2014)
14. Newman, M.E.J.: The structure and function of complex networks. *SIAM Rev.* **45**(2), 167–256 (2003)
15. Newman, M.E.J.: *Networks: An Introduction*. Oxford University Press, Oxford (2010)
16. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. *Phys. Rev. E* **69**(2) (2004)
17. Nicosia, V., Mangioni, G., Carchiolo, V., Malgeri, M.: Extending the definition of modularity to directed graphs with overlapping communities. *J. Stat. Mech. Theor. Exp.* **2009**, P03024 (2008)
18. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**(7043), 814–818 (2005)
19. Plantié, M., Crampes, M.: Survey on social community detection. In: *Social Media Retrieval* (2013)
20. Saracco, F., Di Clemente, R., Gabrielli, A., Squartini, T.: Randomizing bipartite networks: the case of the world trade web. *Sci. Rep.* **5**, 10595 (2015)
21. Saracco, F., Straka, M.J., Clemente, R.D., Gabrielli, A., Caldarelli, G., Squartini, T.: Inferring monopartite projections of bipartite networks: an entropy-based approach. *New J. Phys.* **19**(5), 053022 (2017)
22. Shen, H., Cheng, X., Cai, K., Hu, M.B.: Detect overlapping and hierarchical community structure in networks. *Physica A: Stat. Mech. Appl.* **388**(8), 1706–1712 (2009)
23. Spyropoulou, E., De Bie, T., Boley, M.: Interesting pattern mining in multi-relational data. *Data Min. Knowl. Disc.* **28**(3), 808–849 (2014)
24. Wan, L., Liao, J., Wang, C., Zhu, X.: JCCM: Joint cluster communities on attribute and relationship data in social networks. In: *Proceedings of 5th International Conference on Advanced Data Mining and Applications* (2009)
25. Yang, J., Leskovec, J.: Defining and evaluating network communities based on ground-truth. *Knowl. Inf. Syst.* **42**(1), 181–213 (2015)



Community Detection in a Multi-layer Network Over Social Media

Maham Mobin Sheikh^(✉) and Rauf Ahmed Shams Malick

National University of Computer and Emerging Sciences, Karachi, Pakistan
maham.mobin.sheikh@gmail.com

Abstract. Detection of Communities over the social network, also known as network clustering, has been widely studied in the past few years. The objective of community detection is to identify strongly connected components in a complex network. It reveals how people connect and interact with each other. In the real world, however, a person is engaged in several traits of connections, these connections or social ties carry other different challenges in community detection. More than one trait of connections can be exhibited as a multiplex network that contained itself a collection of multiple interdependent networks, where each network represents a trait of the connections. In this literature, we provide readers with a brief understanding of multilayer networks, community detection methods, and proposed an approach to detect community and its structure using a multi-layer modularity method on the Facebook page. The study also investigates how strong the ties between users and their polarity towards the page over the span of time. The results successfully remove the isolates from the network and built a well-defined structure of the community.

Keywords: Social network · Community structure · Facebook page community

1 Introduction

Social network analysis is rapidly growing in recent years, one of the main reasons is the growing social media platforms. Community detection can be utilized in such disciplines as marketing, information propagation, identifying ethnic groups in society, and so on. As social media platforms are increasing, virtual communities and networks are also expanding and social networks are now have become multilayered. Community detection aims to divide a network into several strongly-connected components. Such subcomponents are formed from sets of similar nodes, and thus can be viewed as a community. If we compare the traditional problem of community detection with a multilayer network then we can conclude that the recent era of mobile phones and social network analyses brings difficulties. Multiple human interactions networks are encapsulated in graphical data such as a person may have a co-worker network at the same time he has a friendship network and will also appear in the social network of online sites. These all networks are interdependent and represent a person's lifetime network which is, in other words, is a temporal network or a complex network, thus identifying a community in

such a network is a challenge and takes a lot of attention from researchers. The objective of this research is to provide a layer-based approach to form a layered network from raw data and find communities over this social network.

Previous research work related to multilayer social networks generally formed the network from directly connected nodes, such as the Facebook network focuses on the friend list of a user, similarly, the LinkedIn network is based on a person's connections. There are very few studies that focus on public pages and posts on social networks. Where users indirectly can form communities and interact with each other using sharing commenting and mentioning others on posts.

We have discovered that there are 60 million active business, political, news channels, and other Facebook pages. People like and share the content of a Facebook page and interact with each other via various activities on pages. We have found that discovering community structure on a public page becomes more difficult as the number of Facebook users and there interactions on pages increases with the increase of features provided by the page. We proposed a method which forms a multilayer network from a given Facebook page and finds indirect communities relation and acquaintanceship within the page.

2 Related Work

Several research works have addressed and proposed community detection methods in a single layer and multilayer network. In this section, we discuss some multilayer community detection methods.

2.1 Community Detection Methods in a Multilayer Network

This study narrows down three main approaches that have been used to extract communities in a multilayer network.

Flattening Approach. In this approach, a multilayer network is first flattened to a single graph network by merging all its layers and then apply a traditional community detection method on that network. Berlingerio et al. [1] proposed an edge count method for community detection, by placing an edge between two vertices if they share one or more layers, these edges then assigned a proper weight depending on their connection. Rocklin and Pinar [2] proposed a predefined community structure to aggregate the weights of edges that come from different layers. The structure is based on the agglomerative clustering technique. Kim, Jungeun et al. [3] proposed a differential flattening method, which combines several layers into a single one such that a single graph layer exhibits structure of the “maximum clustering coefficient”, this method discovers high-quality communities.

Layer-by-Layer or Aggregation Approach. In this approach, each layer of a multi-layer network is processed independently and the resulting solutions are aggregated. The aggregation phase is the key to differentiate between these methods. Tagarelli et al. [4] proposed a modularity-driven ensemble-based approach to find community structure in each layer and then form a structure of community for a multilayer network this structure is known as the Consensus community. This method is fast and obtains a topological community structure. Yuming et al. [5] introduced a “belief propagation algorithm” for community detection in general multilayer networks considering natural label constraints. They apply the Stochastic Block Model (SBM) on each layer for local communities. They consider a multilayer network as a message-passing model and apply a belief propagation algorithm for aggregation. Zhu et al. [6] proposed a cross multi-network community detection method based on the non-negative matrix factorization technique. The algorithm identifies overlapping communities in social networks by matrix decomposition.

Direct or Multilayer Approach. These approaches operated directly on multilayer networks and include clique based methods, random walk based methods, and modularity based methods and label propagation methods. Lucas et al. [7] implement a generalized Louvain method which is an extension of the classic Louvain method. It uses multi-slice modularity and assigns a node-layer tuple separately to each community. Afsarmanesh and Magnani [8] proposed a Multilayer Clique Percolation Method which is the extension of the popular clique percolation method it includes cliques and clique adjacency to ensure the presence of multiple types of edges. Adjacent cliques are assembled to build communities using clique-clique matrix. Zhang et al. [9] proposed the “multilayer edge mixture model” which is derived from a conventional role model that build connection pairs of layers and links probabilities. The hyper model is based on a mixture of edges and weights that reflects their roles in the community detection process.

In the traditional approach, social networks are constructed from directly connected or related nodes, most of the networks were single layer based on one’s friend circle and multilayer networks are based on one’s family, friends, or colleague circles. These networks wouldn’t capture a person’s interactions outside relationships or we can say acquaintanceship of a person through social media activities.

Our proposed method will construct a multilayer network over a Facebook public page to capture the acquaintanceship of a person through the activities and detect community, their structure, and temporal analysis of dynamics on the page.

3 Proposed Work

3.1 Dataset

The Facebook dataset is crawled from “Pakistan Tehreek-e-Insaf (www.facebook.com/PTIOfficial)” public page. On this page only the administration can post information and users can only like, share, comment or reply to a comment. In our database we have two tables one is for post and the other is for comments. The schema of the post is composed

of Author, Time, Text, URL, and post_id. And the schema of comment is composed of profile_id, time, Text, and post_id. Since comments are the most common activity of Facebook so we mainly focus on the comment table (Table 1).

Table 1. Metadata of Facebook page

Data	Measures
Number of users	9457
Number of comments	14196
Number of posts	400
Duration	June 2019 to October 2019

3.2 Proposed Approach

The proposed general framework is represented below. In this approach, we first construct a multilayer network from the dataset which can be represented by a bi-adjacency matrix. This matrix represents all the layers in the network, then using a flattening scheme we transformed the multilayer network into a merged network. From this merged network, we compute a User-User matrix to determine which node should be selected concerning the confidence to be in the acquaintanceship network. In an acquaintanceship network, each node will be part of an overlapping community. A proposed modularity maximization method is applied to acquaintanceship network to detect overlapping communities with positive negative and neutral ties between them (Fig. 1).

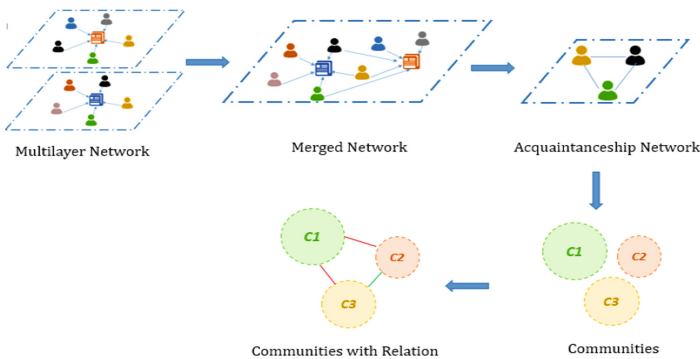


Fig. 1. A simple step by step process of proposed model

3.3 Network Formulation

We considered the multilayer network as graphs where each post represents a layer with a different set of overlapping nodes, a node represents a Facebook user and an edge represents the interaction between a pair of nodes.

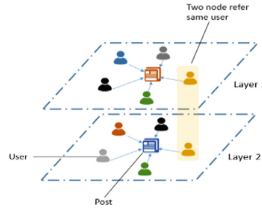


Fig. 2. A toy-example of a multilayer network with two types of interaction among five users

Let P be a set of posts, U be a set of users who have commented on posts and L be a set of layers where a layer represent a single graph containing post and commenters then the multilayer network of the Facebook page is defined as a graph $G = (V, E)$ where G is a multilayer graph, $V \subseteq U \times L \times P$ is the set of nodes or vertices and E is a set of intra-layer edges connecting users to post on the same layer. Layers are not required to contain all users and have a different set of edges. The existence of a user in a layer is represented as a unique node in that layer (Fig. 2).

Defining the Acquaintanceship Network. We defined the community as “If two or more users shared some posts with certain confidence then they are considering as community”. Here shared means that they both have commented on a post. The number of posts users shared is used as a confidence for determining the edge between two users in a community. For detecting the community in this multilayer network we will apply a flattening approach which consists of simplifying the network into a single graph by merging its layers. When a user is existent in two or more layers, this will represent as a single node in the merged layer. Mathematically the merged layer is represented by a bi-adjacency matrix, with one row for each user, one column for each layer/post, and element (i, j) indicating if user i and post j are connected by an edge in the corresponding layer.

$$[B_{i,j}] = \begin{cases} 1, & \text{if user } i \text{ commented on post } j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

This bi-adjacency matrix $B_{i,j}$ is then used with the Matrix transposition method to form a User-User matrix $X_{i,j}$, with one row/column for each user, and element (i, j) indicating the number of shared posts between user i and user j called acquaintanceship weight, denoted as $wt(A_{u_i}^{u_j})$

This User-User matrix is a Gramian Symmetric matrix representing the similarity (or difference) between two users. Matrix $B_{i,j}$ will determine which nodes are selected to be in the acquaintanceship network. The flattening network is then transformed into an acquaintanceship network. In which each node will be a part of an overlapping community.

$$[X_i, j] = B.B^T \quad (2)$$

A Confidence value c will be used to determine the strength and density of the Acquaintanceship network. Users (i,j) who have the score above confidence value will be the part of the Acquaintanceship network. This network can also be defined as a graph $G' = (V', E')$ Where G' is a graph, V' is the set of user nodes having inter-layer edges E' between two users whose $wt(A_{u_i}^{u_j}) \geq c$.

After the formation of the network, we detect communities in this newly formed acquaintanceship network by applying a modularity maximization algorithm Louvain [7]. Modularity measures how densely a network is connected when partitioned into communities. The algorithm divides the graph into clusters called communities and tries to maximize the modularity of a community by placing each node in a different cluster and calculating modularity gain ΔQ for that cluster. It evaluates that how much densely nodes can be connected within a community as compared to how densely they would be in a random network. The gain in modularity obtain by moving u_i in a community C can easily be computed by

$$\Delta Q = \left[\frac{\sum in + 2k_{i,in}}{2m} - \left(\frac{\sum tot + k_i}{2m} \right)^2 \right] - \left[\frac{\sum in}{2m} - \left(\frac{\sum tot}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right] \quad (3)$$

where $\sum in$ is the sum of links inside a community, $\sum tot$ is the sum of links outside the community, k_i is the sum of links of node i , $k_{i,in}$ is the sum of links of node i in community, and m is the total number of links.

Louvain algorithm comprises of two phases, first phase computes the modularity gain ΔQ for all communities if a node moves from one community to other. The next phase is to aggregate all communities which maximize the modularity to form a new graph. We borrow the idea of the Louvain algorithm's first phase which is to maximize the modularity gain. Our proposed algorithm tends to identify communities with signed edges between nodes to further identify relations or ties between them. It will reveal how strongly or weakly a node is tied within the community. It can also be used to predict future relations and the strength of the connection.

The identification of signed edges is based on the polarity of comments given by users on posts. If two users concurrently commented on a post with the same polarity and are members of the same community then they have a strong positive relationship between them. If two users commented on a post with opposite polarity and are member of different communities, then they have a strong negative relationship between them.

The resultant network will be formed by aggregating the communities with signed edges between nodes.

Algorithm: Community Detection with Polar Relation

Input: graph $G = (V, E)$ Output: Aggregated graph $G = (V, R)$

1. $G' \leftarrow \text{CommunityNetwork}(G)$
 2. $C \leftarrow \text{Louvain}(G')$
 3. $R \leftarrow \text{Polarity}(G')$
 4. $G \leftarrow \text{Aggregate graph}$
nodes based on C and make signed edge according to R
-

Procedure: CommunityNetwork

Input: graph $G = (V, E)$ Output: graph $G' = (V', E')$

1. Make bi-adjacency matrix $B_{i,j}$ from equation (1)
 2. Compose User-User matrix $X_{i,j}$ from equation (2)
 3. Set a confidence c
 4. For all $u \in V$
if u_i and u_j has $\text{wt}(A_{u_i}^{u_j}) \geq c$
 $e' \leftarrow \{u_i, u_j\}$
 $V' \leftarrow u_i, u_j$
 $E' \leftarrow e'$
 5. return $G' = (V', E')$
-

Procedure: Louvain

Input: graph $G' = (V', E')$ Output: list $C: V' \rightarrow [1: N]$

1. $C \leftarrow V'$
 2. for all $u \in V'$
Compute $\Delta Q(C)$ from equation (3)
Move u to adjacent communities and compute ΔQ .
 $C_{\text{new}} \leftarrow \text{argmax } \Delta Q(C) \quad C \in \text{adj_comm}(u)$ - set of adjacent communities of u
If $\Delta Q(C_{\text{new}}) > 0$
 $C[v] \leftarrow C_{\text{new}}$
 3. return C
-

Procedure: Polarity

Input: graph $G' = (V', E')$ Output: list $R: E' \rightarrow [+,-,0]$

1. for all $u \in V'$
 $\text{Polarity} \leftarrow 0$
 $\text{Raw} \leftarrow \text{Concatenate (all comment attribute of } u)$
 $\text{Tokenize}(\text{Raw})$
 $\text{PartOfSpeechTagger}(\text{tokens})$
 $\text{Polarity} \leftarrow \text{argmax } P(\text{Sign}|\text{tokens}) \quad \text{Sign} \in [+,-,0]$
2. if $(u_i$ and u_j has same polarities)
 $E' \leftarrow \{u_i, u_j, +\}$
Else if $(u_i$ and u_j has opposite polarities)
 $E' \leftarrow \{u_i, u_j, -\}$
Else $E' \leftarrow \{u_i, u_j, 0\}$
3. return E'

Evaluation Metrics. There are two common evaluation metrics of community detection the first is accuracy if the actual member of the community is given. And second

is modularity. In social networks, it is challenging to identify a member of the community. In this research we use two matrices to evaluate the community structure first is modularity and the second is similarity we find similar users based on their polarity of comments. The similarity score can be defined as:

$$\text{Similarity score} = \frac{\sum_0^n \text{Sim}}{N} \text{ where } N \text{ is total no of nodes} \quad (4)$$

$$\text{Sim} = \frac{\text{similar nodes in community}}{\text{total nodes in community}} \quad (5)$$

4 Results

In this section, we will present the analysis of the results of our proposed method on the multilayer network of Facebook page graph. The single layer is composed of a single post and users who have commented on that post.

4.1 Community Detection in Multilayer Network

Our multilayer network consists of 400 layers and an extra step of flattening the network into a Merged Network. This Merged Network is transformed into an acquaintanceship network of Users having a confidence score c . The confidence score will tell the strength of ties between users. Figure 3 shows the Merged Network, Figs. 4, 5 and 6 shows the community structure of acquaintanceship network when $c = 2, 5,$ and 8 respectively. The adopted method identified global communities and removed isolated nodes from the network. The results show that for a lower confidence score, communities are denser and modular as compare to the higher confidence score but have weak ties whereas high confidence score communities have more strengthen ties between similar users and show negative relations between communities as shown in Fig. 6. Identified community structure information is present in Table 2. It is observed that less modular communities have the most similar and strengthen ties.

Table 2. Effect of confidence on community structure

Confidence	Nodes	Communities	Modularity	Similarity
3	192	11	0.43	0.65
5	37	4	0.35	0.8
8	8	2	0.22	1

Table 3 provides the comparison of flattening algorithms characteristics with the proposed algorithm on the bases of C1: Gives importance to layers, C2: Layer relevance weights, C3: Gives importance to actors, C4: Removes isolates, C5: Predefine community structure, C6: Predefine number of communities, C7: Strengthen community members, C8: Find similarity of members within a community.

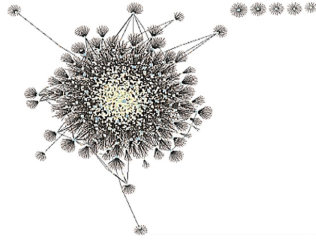


Fig. 3. The merged network formed by flattening 400 layers

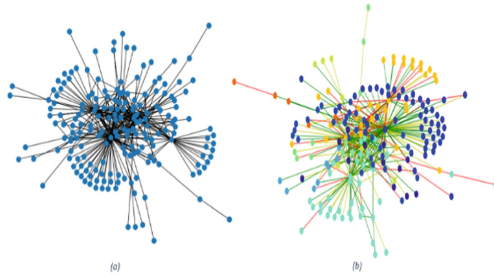


Fig. 4. Community structure with tie strength in acquaintanceship network with $c = 3$ (a) Acquaintanceship network. (b) Communities with relations

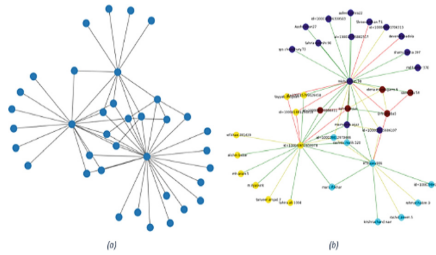


Fig. 5. Community structure with tie strength in acquaintanceship network with $c = 5$ (a) Acquaintanceship network. (b) Communities with relations

Table 3. Comparison of flattening algorithms characteristics

Algorithm (weighted flattening)	C1	C2	C3	C4	C5	C6	C7	C8	Ref
Edge count	X	X	X	X	X	X	X	X	[1]
Aggregated clusters	X	X	X	X	✓	✓	X	X	[2]
Differential flattening	✓	✓	X	X	X	X	X	X	[3]
Proposed method	X	X	✓	✓	X	X	✓	✓	

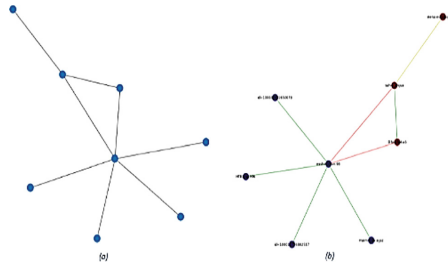


Fig. 6. Community structure with tie strength in acquaintanceship network with $c = 8$ (a) Acquaintanceship network. (b) Communities with relations

4.2 Social Network Analysis of Merged User Graph

We have performed some basic analyses on the merged graph. Table 4 shows the network statistics of the merged graph. Figure 7 shows the Betweenness Centrality.

Table 4. Facebook page network statistics

Data	Measure
Nodes	9227
Edges	314061
Avg. degree	68.07
Avg. path length	2.68
Clustering coefficient	0.901
Network diameter	5
Network radius	3

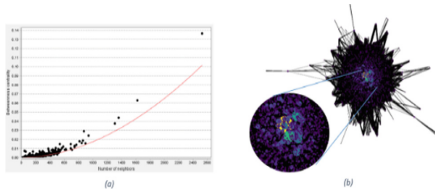


Fig. 7. (a) Betweenness centrality distribution. (b) Betweenness centrality graph

Figure 8 shows the impact of the removal of ten highly centric nodes vs ten random nodes on the overall connectedness of the network. From the Figure, we can conclude that by removing centric nodes the network will disconnect sooner but removing random nodes has little to no effect on network connectedness.

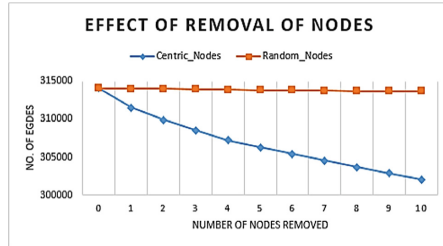


Fig. 8. Impact of removal of nodes on overall connectedness of the network

Similarly, Fig. 9 shows the impact of the removal of ten highly centric nodes and ten random nodes on average path length. We can see that as the number of highly centric nodes decrease average path length increases, but the removal of random nodes does not affect average path length. From Figs. 8 and 9 we can conclude that the removal of random nodes has no effect on the network state but removing centric nodes increased the diameter of the network and decreased network centralization and average degree.

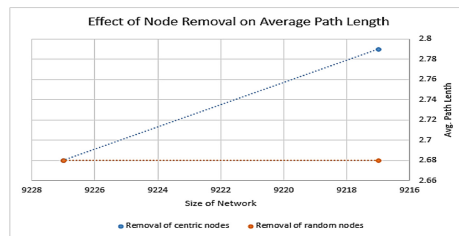


Fig. 9. Impact of removal of nodes on the average path length of the network

4.3 Temporal Analysis of User’s Polarity in Network

We have done a temporal analysis of the user’s comment in the network to identify the polarity of ties over time and to investigate the overall consistency of the user’s opinion towards the page.

In Fig. 10 sunburst chart’s each ring depicts a week, the chart shows that after 2 weeks there is a transition in the polarity of a few peoples. That’s mean that a small number of peoples has changed their polarity of comments on the page. Most of the people have consistency in their comments.

To investigate the overall polarity transition of the user’s comment with respect to time, we plot graphs for positive, negative, and neutral users. Figure 11 shows

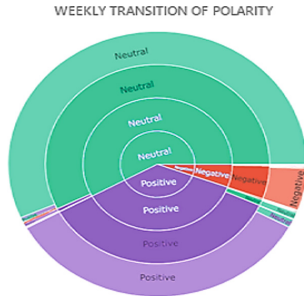


Fig. 10. The weekly transition of the polarity of user comments

graphs from which we can conclude that by the passage of time positive and negative commenters are decreasing whereas the number of neutral commenters increases.

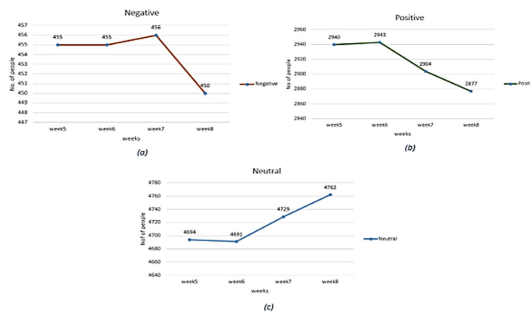


Fig. 11. Commenter’s polarity over the time (a) Negative commenters graph (b) positive commenters graph (c) Neutral commenters graph

5 Conclusion

In this research, we have developed a flattening technique to identify Global communities in a multilayer network constructed from the Facebook public page. We consider page posts as layers and commenters as members of an overlapping community. The algorithm successfully removed the isolated nodes and construct the community over highly centric nodes. The resultant communities are modular and revolve around the highly centric users, the algorithm also identified the strength of ties between users. Such community structures can be used to find the active influential nodes over the public page.

References

1. Berlingerio, M., Coscia, M., Giannotti, F: Finding and characterizing communities in multidimensional networks. In: 2011 International Conference on advances in social networks analysis and mining. IEEE, pp. 490–494 (2011, July)

2. Rocklin, M., Pinar, A.: On clustering on graphs with multiple edge types. *Internet Math.* **9**(1), 82–112 (2013)
3. Kim, J., Lee, J.G., Lim, S.: Differential flattening: A novel framework for community detection in multilayer graphs. *ACM Trans. Intell. Syst. Technol. (TIST)* **8**(2), 1–23 (2016)
4. Tagarelli, A., Amelio, A., Gullo, F.: Ensemble-based community detection in multilayer networks. *Data Min. Knowl. Disc.* **31**(5), 1506–1543 (2017). <https://doi.org/10.1007/s10618-017-0528-8>
5. Huang, Y., Krim, H., Panahi, A., Dai, L.: Community detection and improved detectability in multiplex networks. *IEEE Trans. Netw. Sci. Eng.* **7**(3), 1697–1709 (2020). Electronic ISSN: 2327-4697. <https://doi.org/10.1109/TNSE.2019.2949036>
6. Zhu, Z., Zhou, T., Jia, C., Liu, J., Cao, J.: Community detection across multiple social networks based on overlapping users. arXiv preprint [arXiv:1909.09007](https://arxiv.org/abs/1909.09007) (2019)
7. Jeub, L.G.S., Bazzi, M., Jutla, I.S., Mucha, P.J.: “A generalized Louvain method for community detection implemented in MATLAB” (2011–2019)
8. Afsarmanesh, N., Magnani, M.: Finding overlapping communities in multiplex networks. arXiv preprint [arXiv:1602.03746](https://arxiv.org/abs/1602.03746) (2018)
9. Zhang, H., Wang, C.D., Lai, J.H., Philip, S.Y.: Community detection using a multilayer edge mixture model. *Knowl. Inf. Syst.* **60**(2), 757–779 (2019)



Using Preference Intensity for Detecting Network Communities

József Dombi and Sakshi Dhama^(✉)

University of Szeged, Szeged 6720, Hungary
{dombi,sakshi}@inf.u-szeged.hu
<http://www.inf.u-szeged.hu/~dombi/>

Abstract. In a real-world network, overlapping structures are essential for understanding the community. In many different situations, a node may join or leave, and this defines sub-communities of varying size. In this paper, we propose a preference implication based-method for generating overlapping structures based on a local function optimization approach. We introduce some parameters in our novel method to design the communities according to a threshold. This method allows us to control the size and number of these overlapping regions. The ν will enable us to design the sub-communities. This framework can easily detect communities in a scale-free network case. We set our experiments using artificial and real network data with a size between ≈ 15 to ≈ 10000 . In our findings, we found a good relationship between ν and overlapping nodes in communities. We control our procedure using α parameter as well. We can say that the preference is stronger when ν is greater than 0.5, and a value of α between 0.20 and 0.80. The third parameter δ , which controls the intensity of community membership, defines the degree of relationship of a node to a community. The communities detected by the preference implication method obey a power law in the community size distribution.

Keywords: Preference relations · Overlapping communities · Power law networks · Continuous value logic · Community detection

1 Introduction

In many real-life applications, we find many high-value relationships among the data. Networks are mathematical structures which consist of vertices and edges. Sometimes inter-related data is modelled as graphs to study the association and other essential features in data. The links of a node contribute to the control dynamics of a network [1]. Finding community on the networks is one of the critical problems to study. A real network consists of nodes that may belong to more than one community [2–4]. The overlapping regions are common in networks where some nodes can exhibit properties of more than one community [5, 6]. Community detection in networks with more than one membership is of

great interest as it resembles more closely the real-world networks [2]. For example, in a protein complex network, a large number of proteins may belong to many protein complexes at the same time [7]. The identification of these community structures can provide a solution for many risky situations. For example, to control the community infection earlier and in the time of pandemic like the current one is of great interest. Many community detection methods have been developed in the past three decades, and we present a summary of some. Some algorithms use a local function to characterize the densely connected group of nodes [8–22]. Lancichinetti uses the local function optimization approach in the (LFR) overlapping community benchmark [23]. LFR introduces a fitness function for the definition of a community, as shown in Eq. 1. The random seed nodes from the network form the community until the fitness function in Eq. 1 is locally maximal. In the OSLOM method, the local optimization of the fitness function is used, which determines the statistical significance of clusters for random fluctuations [24]. First, it identifies the relevant cluster until the local fitness function converges. Then an internal analysis of these clusters is performed of their union. Lastly, it identifies the hierarchical structure of these clusters. This method gives a comparable performance with those of other existing algorithms on synthetic networks. The main advantage of this method is that it can be used to improve the clusters generated by different algorithms

2 A New Approach for Community Detection

One of the hypotheses in the definition of the community asserts that the community is a locally dense connected subgraph in a network. In the case of the LFR benchmark, the weak community definition is used to define the fitness function of the community and the fitness function for a community is:

$$f_{\mathcal{G}} = \frac{K_{in}^{\mathcal{G}}}{(K_{in}^{\mathcal{G}} + K_{out}^{\mathcal{G}})^{\alpha_1}}, \quad (1)$$

where \mathcal{G} is the subgraph or community, $K_{in}^{\mathcal{G}}$ is the total number of internal links, $K_{out}^{\mathcal{G}}$ is the total number of links of each member relative to the total graph and α_1 is a positive real-valued parameter which controls the size of communities. In the method used by LFR, new node ‘a’ is added in the community if

$$f_{\mathcal{G}'} > f_{\mathcal{G}}, \quad (2)$$

where $f_{\mathcal{G}}$ is the fitness function of community prior to addition of node a and $f_{\mathcal{G}'}$ is the fitness of community after the addition of node a .

$$f_{\mathcal{G}'} = \frac{K_{in}^{\mathcal{G}'}}{(K_{in}^{\mathcal{G}'} + K_{out}^{\mathcal{G}'})^{\alpha_1}} \quad (3)$$

1. In this method, the first limitation is that the overlap regions may belong to more than one community, which is difficult to decide based on Eq. 2.

2. In a more practical real-world situation, every community has a threshold for membership criteria. In the LFR benchmark, the inequality operator is limited, as there are no criteria to control the strength of a community relative to a threshold.
3. This method also generated a fixed community membership for all the overlapping nodes.

To overcome these limitations of Eq. 2, we propose the so-called preference implication-based method, to control the threshold. The user can control the strength of the community by changing the threshold value. For example, social networks differ from other networks [25]. In a social network, when a new network community begins to recruit members, the joining threshold to become a member of this community is very low. After a certain period, the membership threshold increases as the network community has now matured.

3 Preference Relations

This new method based on preferences can be used to define overlap communities in networks and also be used to create benchmark communities. The preference relation has the monotone property, and here we define the preference implication which can be used to make multi-criteria decisions [26, 27]. The preference relation $P_\nu^{(\alpha)}(x, y)$ shows how true is $(x < y)$ sometimes, which in our case also indicates how strong the community is. Here, $x = f_G$ and $y = f_{G'}$

$$P_\nu^{(\alpha)}(x, y) \text{ where } x < y \text{ and } x, y, \nu \in (0, 1) \quad (4)$$

$$P_\alpha^\nu(x, y) = \text{degree}(x < y) \quad (5)$$

$P(x, y) \in (0, 1)$ and in Boolean algebra $P(0, 1) = 1$ and $P(1, 0) = 0$. In Table 1 the domain of all the parameters we have used in the preference-based method is explained. For example, the preference value based relation of truth of $(x < y)$ have these three calculated possible values,

1. $P(6 < 9) = 0.9$
2. $P(6 < 6) = 0.5$
3. $P(9 < 6) = 0.1$

$P_\nu^{(\alpha)}(x, y) > \nu$ if and only if $x < y$. Now, let us assume that the threshold ν is 0.5 and that the sharpness parameter α is 1.

Case 1. $P_\nu^{(\alpha)}(6, 9)$ The truth value of statement $(6 < 9)$ is 0.9, which greater than ν as $0.9 > 0.5$. Hence, we establish the truth statement $6 < 9$ (using preference relation) with a strength quite greater than ν (threshold).

Case 2. $P_\nu^{(\alpha)}(6, 6)$ For the truth value of statement $(6 < 6)$ is 0.5 which is just the threshold value. So in this case $6 < 6$ is a weak statement as it is at the threshold, but still it establishes the truth of the statement.

Table 1. The range of the parameters values in the preference relation

Parameters	Domain
Preference Relation	$P_\nu^{(\alpha)}(x, y) \in (0, 1)$
Threshold	$\nu \in (0, 1)$
Sharpness Parameter	$\alpha \in (0, 1)$
$x = f_{\mathcal{G}}$	$f_{\mathcal{G}} \in (0, 1)$
$y = f_{\mathcal{G}'}$	$f_{\mathcal{G}'} \in (0, 1)$
$\delta \in \text{deltaset}$	$(\text{deltaset}) \in (0, 1)$

Case 3. $P_\nu^{(\alpha)}(9, 6)$ The truth value of statement $(9 < 6)$ is 0.1, which is less than ν as $0.5 > 0.1$. So, it is a very weak statement or in other words it is a false statement. And hence, $9 < 6$ is not true a statement.

Here the sharpness threshold is defined by α and ν is the threshold used for comparing the truth values. The intensity of the preference is controlled by the parameter of this function. The parameter is $\nu \in (0, 1)$ and f is a generator of a strict t norm. The preference implication in pliant logic form is [26]:

$$P_\nu^{(\alpha)} = \begin{cases} 1, & \text{if } (x, y) \in (0, 0), (1, 1) \\ f^{-1}\left(f(\nu) \frac{f(y)}{f(x)}\right)^\alpha, & \text{otherwise} \end{cases} \tag{6}$$

We can also define our own function in the Eq. 6. We define a special function for our purpose which has the monotonic property [28, 29].

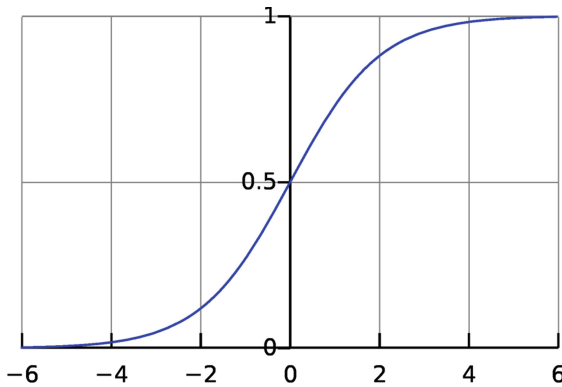


Fig. 1. The preference relation sigmoid function

Using the Dombi operator for the preference relation in f_G , we get

$$P_\nu^{(\alpha)}(x, y) = \frac{1}{1 + \frac{1-\nu}{\nu} \left(\frac{1-y}{y} \frac{x}{1-x} \right)^\alpha} \quad (7)$$

$$P_\nu^{(\alpha)}(x, y) > \nu \text{ if and only if } x < y, \quad (8)$$

$$P_\nu^{(\alpha)}(x, y) = \begin{cases} > \frac{1}{2}, & \text{if } y > x \\ = \frac{1}{2}, & \text{if } x = y \\ < \frac{1}{2}, & \text{if } x > y \end{cases} \quad (9)$$

Table 2. Preference rule table: rule for the new node addition in the subgraph based on the threshold ν value and preference value.

Preference value $P_\nu^{(\alpha)}(x, y)$	Fitness function rule	Decision of a new node addition
$0 < P_\nu^{(\alpha)}(x, y) < 0.5$ $P_\nu^{(\alpha)}(x, y) > \delta$	$f_G > f_{G'}$	Not desirable for a community membership but it can have a very weak overlapping membership
$P_\nu^{(\alpha)}(x, y) = 0.5$ $P_\nu^{(\alpha)}(x, y) > \delta$	$f_G = f_{G'}$	Desirable for Community membership and it can have a weak overlapping membership
$1 > P_\nu^{(\alpha)}(x, y) > 0.5$ $P_\nu^{(\alpha)}(x, y) > \delta$	$f_G < f_{G'}$	Strong community membership and it can have a strong overlapping membership

This method allows us to control the size and number of these overlapping regions. The threshold parameter ν of preference allows us to design the communities according to our requirement of the strength of a community. The value of ν is desirable when $\nu > 0.5$, as the Dombi operator system is a sigmoid function (as in Eq. 9). The graphical form of the sigmoid function is shown in Fig. 1. Table 2 above lists the preference-based rules in different scenarios and its comparison with different values of ν . Also, the case of strong membership occurs when the threshold value is greater than 0.5.

4 Preference Relation Properties

Theorem 1. *The necessary and sufficient conditions for satisfying all the four distributivity equations are [29]:*

1. The conjunction and disjunction are weighted operators.
2. Negation is a strong negation.
3. The De Morgan laws are valid for the above triple.
4. The implication is a fuzzy implication which is continuous except for the points $(0, 0)$ and $(1, 1)$.
5. The law of contrapositive is valid. And these conditions can only be satisfied if the operators are elements of a pliant system and the implication is a preference implication. That is,

$$c(x, y) = f^{-1}(uf(x) + vf(y)),$$

$$d(x, y) = f^{-1}\left(\frac{f(x)f(y)}{vf(x)+uf(y)}\right),$$

$$\eta(x) = f^{-1}\left(\frac{f^2(v)}{f(x)}\right),$$

$$\begin{cases} 1, & \text{if } (x, y) \in (0, 0), (1, 1), \\ f^{-1}\left(f(\nu)\frac{f(y)}{f(x)}\right), & \text{otherwise,} \end{cases}$$

for all $x, y \in [0, 1]$ where $u, v \in (0, \infty)$ and $\nu \in (0, 1)$

Definition 1. For $x, y \in [0, 1]$, $P(x, y)$ has the reciprocity property when

$$P(x, y) + P(y, x) = 1 \tag{10}$$

Definition 2. A preference relation p is multiplicative transitive if

$$\frac{p(x, y)p(y, z)}{p(y, z)p(z, y)} = \frac{p(x, z)}{p(z, x)} \tag{11}$$

for all x, y, z in $[0, 1]$ and the above formula is well defined.

Note 1. We define this special function for our purpose, and it has the monotonic property. We are using the Dombi operator for the preference relation in Eq. 6.

Definition 3. A preference implication p is reciprocal if

$$p(x, y) + p(y, z) = 1 \quad x, y \in [0, 1] \tag{12}$$

Here, we will prove that for preference implication,

$$P_\nu^{(\alpha)}(x, y) > \nu \quad \text{if and only if } x < y \tag{13}$$

Proof: As we know the form of preference implication is:

$$P_\nu^{(\alpha)}(x, y) \text{ where } x < y \text{ and } x, y, \nu \in (0, 1). \tag{14}$$

Using the Dombi operator for the preference relation from Eq. 7, and noting 1, we get

$$P_\nu^{(\alpha)}(x, y) = \frac{1}{1 + \frac{1-\nu}{\nu} \left(\frac{1-y}{y} \frac{x}{1-x}\right)^\alpha}. \tag{15}$$

Now, from Eq. 7 we get the expression given below,

$$\frac{1}{1 + \frac{1-\nu}{\nu} \left(\frac{1-y}{y} \frac{x}{1-x} \right)^\alpha} > \nu. \quad (16)$$

Taking the reciprocal of the LHS and RHS, we get

$$\frac{1-\nu}{\nu} \left(\frac{1-y}{y} \frac{x}{1-x} \right)^\alpha < \frac{1-\nu}{\nu}. \quad (17)$$

Subtracting (17) from 1, we get

$$\left(\frac{1-y}{y} \frac{x}{1-x} \right)^\alpha < 1. \quad (18)$$

After cross multiplication of the above term we get the following reduced form:

$$(1-y)x < y(1-x), \quad (19)$$

$$x - xy < y - xy. \quad (20)$$

Cancelling the common term $-xy$ on each side of the equation we get,

$$x < y, \quad (21)$$

and hence it is proved. Therefore,

$$P_\nu^{(\alpha)}(x, y) > \nu \text{ if and only if } x < y. \quad (22)$$

Commutative Property. Here x is the fitness value of the sub-graph before the addition of a new node, and y is the fitness value of the sub-graph after the addition of a new node. Here, n and o for simplicity denote k_{in} and k_{out} , respectively.

As defined above, we know that

$$x = \frac{k_{in}}{(k_{in} + k_{out})^{\alpha_1}} \quad (23)$$

For simplicity we choose $\alpha_1 = 1$, and we get

$$x = \frac{k_{in}}{k_{in} + k_{out}}. \quad (24)$$

Now, taking the reciprocal of x and subtracting 1 from it, we get

$$\frac{1-x}{x} = \frac{k_{out}}{k_{in}}. \quad (25)$$

Taking reciprocal of the previous expression we get,

$$\frac{x}{1-x} = \frac{k_{in}}{k_{out}}. \tag{26}$$

As defined above in commutative property we get

$$\frac{k_{in}}{k_{out}} = \frac{n}{o}. \tag{27}$$

Similarly, for y when $\alpha_1 = 1$ we get,

$$y = \frac{k'_{in}}{k'_{in} + k'_{out}}. \tag{28}$$

Now we repeat the same steps for y as we did for x ,

Taking reciprocal and subtracting and taking reciprocal again, we get

$$\frac{1-y}{y} = \frac{k'_{in}}{k'_{out}}. \tag{29}$$

Also, in another notation, we get

$$\frac{k'_{in}}{k'_{out}} = \frac{o'}{n'}. \tag{30}$$

From 7 we get,

$$P_{\nu}^{(\alpha)}(x, y) = \frac{1}{1 + \frac{1-\nu}{\nu} \left(\frac{1-y}{y} \frac{x}{1-x} \right)^{\alpha}}. \tag{31}$$

We introduce the threshold δ and from 8 we get,

$$P_{\nu}^{(\alpha)}(x, y) > \delta. \tag{32}$$

Therefore,

$$\frac{1}{1 + \left(\frac{o'}{n'} \frac{n}{o} \right)^{\alpha}} > \delta.$$

Taking the reciprocal, we get

$$1 + \left(\frac{o'}{n'} \frac{n}{o} \right)^{\alpha} < \frac{1}{\delta}, \tag{33}$$

$$\left(\frac{o'}{n'} \right) \left(\frac{n}{o} \right) < \left(\frac{1}{\delta} - 1 \right)^{\frac{1}{\alpha}}. \tag{34}$$

Let us assume the RHS of the above expression is k , Then

$$\left(\frac{1}{\delta} - 1\right)^{\frac{1}{\alpha}} = k. \quad (35)$$

And we get,

$$\left(\frac{o'}{n'}\right)\left(\frac{n}{o}\right) < k. \quad (36)$$

Taking the log on both sides of above expression, we get

$$\ln(o') - \ln(o) + \ln(n) - \ln(n') < \ln(k). \quad (37)$$

$$\Delta o - \Delta n < k', \text{ where } k' = \frac{1}{\alpha}(\ln(1 - \delta) - \ln \delta). \quad (38)$$

When a new member is added to the community, and there is an increase in the number of internal links of the community, and also a comparatively small increase in the number of external links. We calculate this difference on a logarithmic scale. This increase is directly related to k , where k is $\left(\frac{1}{\delta} - 1\right)^{\frac{1}{\alpha}}$. This term denotes the strict threshold for the addition of a new member to the community.

5 Framework of the Preference-Based Method

A brief preliminary version of the algorithm was described in our paper [34]. In our framework, the preference-based method works in the following way. We start with the number of community nc to be detected and graph G as input in the main algorithm.

1. We start the process of community creating by randomly selecting seed nodes for each community c . Centrality measures such as PageRank, betweenness, and other centralities can also be used to determine the seed nodes. Each community c is represented by \mathcal{G} , which is a subgraph with one node and one virtual edge.
2. We find all the neighbours of the community. To make the selection of best neighbour to add in the community, we create a new subgraph $\mathcal{G}'_i = \mathcal{G} + i$ corresponding to each neighbour i . Now, we have to choose the best subgraph out of this list; i.e. the best next node or nodes to be added in \mathcal{G} to improve the local fitness function of the community. We also consider the strength of community denoted by δ when making this selection.
3. We create a preference list corresponding to each subgraph of neighbouring node i of \mathcal{G} . If the preference value of any subgraph is greater than the δ parameter value, then these nodes are included in the community.

Algorithm 1: Main Function

```

Input:  $G, nc$  /*  $G$ : Graph on which community detection is to be
        performed,  $nc$ : number of communities */
Output:  $L$  /* List of communities detected on the Graph, i.e. each
        index of  $L$  has one community subgraph */
1 Function Calculate( $i$ ):
2   foreach  $i \in \text{neighborhood of } c$  do
3     /* Select next node using Preference */
4      $Y \leftarrow \text{Fitness}(K_{in}, K_{out})$  /*  $\mathcal{G}'_i = \mathcal{G} + i$  */
5      $\text{PreferenceListof } c_i \leftarrow \text{Preference}(X, Y)$ 
6   return  $\text{PreferenceListof } c_i$ 
7 End Function
8 Function Main( $G, nc$ ):
9    $L \leftarrow \text{head}$  /* Each community is initialised with one node, which
        is the head node */
10   $\text{First\_itr} \leftarrow 1$  /* Flag for first iteration */
11  foreach  $c \in L$  do
12     $\text{delta} \leftarrow (0.1, 0.2, 0.3, \dots, 0.9)$ 
13    /* Initialize threshold */
14     $\mathcal{G} \leftarrow L[c]$ 
15    while  $\mathcal{G}' > \mathcal{G}$  or  $\text{First\_itr}$  /* Stopping condition */
16    do
17       $\mathcal{G} \leftarrow \mathcal{G}'$ 
18       $X \leftarrow \text{Fitness}(K_{in}, K_{out})$  /* Fitness of  $\mathcal{G}$  */
19       $\text{PreferenceListof } c_i \leftarrow \text{Calculate}(i)$ 
20      foreach  $i$  in  $\text{PreferenceListof } c$  do
21        if  $\text{PreferenceListof } c_i > \delta$  then
22          /*  $i^{\text{th}}$  node is member of the community  $c$  */
23           $\mathcal{G}' \leftarrow \mathcal{G}_i$ 
24           $X \leftarrow \text{Fitness}(K_{in}, K_{out})$  /* update the new  $X$  which
                has community with new members */
25        end if
26       $L[c] \leftarrow \mathcal{G}$ 
27    End while
28  return  $L$ 

```

$\mathcal{G}' \leftarrow \sum \mathcal{G}_i$: a new subgraph which includes nodes with a preference value greater than δ .

4. The process is repeated from step 4 using a while loop until it satisfies the stopping criteria.

Algorithm 2: Function for Preference Relation Implication value

```

1 Input:  $F_G, F_{G'}$  /* Fitness of subgraph  $F_G$ , Fitness of the
   subgraph  $F_{G'}$  */
   Output:  $P$  /* Preference value of the subgraph */
2 Function Main( $F_G, F_{G'}$ ):
3    $alpha \leftarrow 1$ ;  $X \leftarrow Normalize(F_G)$ ;  $Y \leftarrow Normalize(F_{G'})$ ; if
   ( $X > 0 \ \&\& \ X < 1 \ \&\& \ Y > 0 \ \&\& \ Y < 1$ ) then
4      $P \leftarrow \frac{1}{1 + \frac{1-\nu}{\nu} \left( \frac{1-Y}{Y} \frac{X}{1-X} \right)^\alpha}$ ;
5   end
6   else
7      $P \leftarrow null$ ;
8   end
9   return  $P$ 
10 End Function

```

Algorithm 3: Fitness Function to calculate fitness of subgraph

```

Input:  $K_{in}, K_{out}$  /* In-degree, Out-degree of subgraph */
Output:  $f$  /* Fitness value of the subgraph */
1 Function Fitness( $K_{in}, K_{out}$ ):
2    $alp \leftarrow 1$ 
3    $f \leftarrow \frac{K_{in}}{(K_{in} + K_{out})^{alp}}$ 
4   return  $f$ 

```

6 Experiments and Results

The key parameters of our algorithm are ν and α . We selected artificial and real networks to test our algorithm [30–32]. For the artificial network, we generated different sizes of networks from the LFR benchmark. The real and artificial networks we selected for tests have quite similar characteristics in terms of the diameter, density, transitivity, maximum and average degree with respect to size. To test the parameters, we chose small-sized networks. Based on our analysis of the ν and α we have generated communities for $\nu \in [0.3, 0.8]$ and $\alpha \in (0.05, 0.95)$. For a larger network, we show the community size distribution in (B) of Fig. 2. The community size distribution is different for different values of ν , but it has a similar value for community sizes with the same threshold. For a smaller value of $\nu < 0.5$, smaller size communities are formed, and they all have similar sizes. However, for values greater than 0.5, larger communities sizes are observed. When the number of seeds for the community is increased, it increases the number of overlapping nodes, but there was little change in the community size behaviour as we used the same threshold for all the seeds. In the case of larger networks, the behaviour power-law is much more visible, as can be seen in Fig. 3 from the shape of violins and in Fig. 3. Community detection was

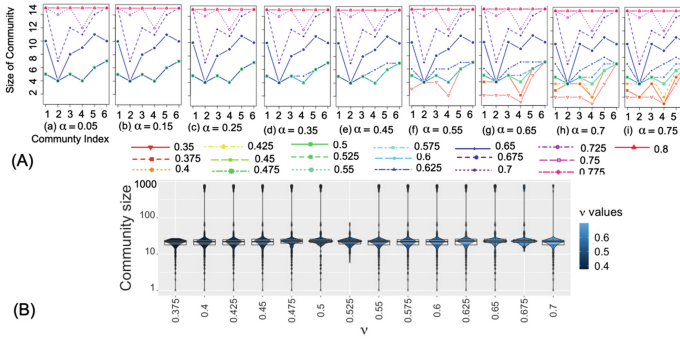


Fig. 2. In (A) of Fig. 2 from the LFR benchmark the statistics of network that we used are $N = 15, k = 3, maxk = 5, \nu = 0.2, t_1 = 2, t_2 = 1, minc = 3, maxc = 5, on = 5, om = 2$. The value of α is tested for $\alpha \in [0.05, 0.8]$. For each α the ν is tested for $\nu \in [0.35, 0.8]$. The initial seed nodes were the same for all the runs. The value of number of communities to be created is the same in all the cases; i.e. $nc = 6$. The delta set is same in each cases with ν and α with the values $[0.24, 0.95]$. The Community size distribution is shown in (B) of Fig. 2 for a fixed value of ($\alpha = 0.75$) and varying value of $\nu \in [0.375, 0.70]$. The size of network = 1000, Average degree $k = 150$, Maximum degree $K_{max} = 200$, Mixing parameter $\mu = 0.1, t = 1, t_2 = 1$. The initial seed node and threshold δ are the same for each value of ν .

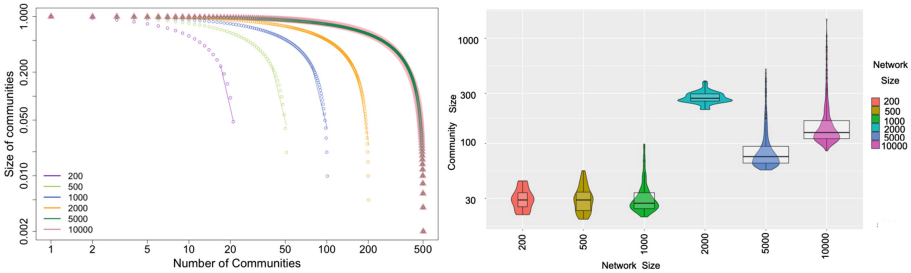


Fig. 3. The figure on right shows the violin plot of a community size distribution for 10% of the seed nodes on a large networks with a size from 200 to 10000. The value of parameters $deltaset = 0.3, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, \alpha = 1$ and $\nu = 0.6$ are the same in every case. The figure on the left shows a power law distribution for a community size of artificial networks with a size from 200 to 10000. The parameter values are the same as those in the violin plot on the right.

performed using different seed size for 5%, 10%, 20% and 30% of the network. We observed a similar behavior for violins in all the cases and they have the characteristics of a power law.

7 Conclusions and Future Work

We conducted our tests on artificial and real networks and found the method promising for controlling the overlapping structures. The parameters δ and α

can be useful to control the overlapping nodes. The delta set is vital for deciding the threshold for the community strength. Different seed sizes were used for the community detection method, and they had a good performance and gave similar results. The detected communities size follow the power-law when tested on different artificial networks with a different seed size ranging from ten to thirty per cent of the network size. The preference implication provides a new way of analysing the creation of overlapping communities in networks for the algorithms which employ a local function optimisation approach for the detection of overlapping communities in the network. Robustness is an important property for many networks [33]. For networks which are not robust, our method of community detection may be useful as it preserves the original structure of the graph when the algorithm terminates [34]. We showed that the algorithm could be helpful for analysing artificial and real networks, it has a good performance, and we calculated the results to demonstrate the effectiveness of the preference implication method. In the future, we intend to include the map-reduce framework for faster implementation of this method. If the network is complex, then the run time increases, and it becomes difficult to collect when we have very large networks. Optimising the code of the current parallelized algorithm using a map-reduce function would be useful for making the algorithm scalable, and it would be beneficial in real-time decentralised networks.

Acknowledgements. We gratefully thank the University of Szeged for providing financial support for this conference.

References

1. Nepusz, T., Vicsek, T.: Controlling edge dynamics in complex networks. *Nat. Phys.* **8**(7), 568 (2012)
2. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**(7043), 814 (2005)
3. Newman, M.E.: The structure and function of complex networks. *SIAM Rev.* **45**(2), 167–256 (2003)
4. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks **99**(12), 7821–7826 (2002)
5. Barabási, A.L.: *Network Science*. Cambridge University Press (2016). <http://networksciencebook.com/>
6. Gregory, S.: Fuzzy overlapping communities in networks. *J. Stat. Mech. Theor. Exp.* **2011**(2), P02017 (2011)
7. Gavin, A.C., Bösch, M., Krause, R., Grandi, P., Marzioch, M., Bauer, A., Schultz, J., Rick, J.M., Michon, A.M., Cruciat, C.M.: Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* **415**(6868), 141 (2002)
8. Baumes, J., Goldberg, M.K., Krishnamoorthy, M.S., Magdon-Ismael, M., Preston, N.: Finding communities by clustering a graph into overlapping subgraphs. In: *IADIS AC*, pp. 97–104 (2005)
9. Derényi, I., Palla, G., Vicsek, T.: Clique percolation in random networks. *Phys. Rev. Lett.* **94**(16), 160202 (2005)

10. Gulbahce, N., Lehmann, S.: The art of community detection. *BioEssays* **30**(10), 934–938 (2008)
11. Kelley, S.: The existence and discovery of overlapping communities in large-scale networks. Ph.D. thesis, Rensselaer Polytechnic Institute (2009)
12. Kim, J., Wilhelm, T.: What is a complex graph? *Phys. A* **387**(11), 2637–2652 (2008)
13. Li, H.J., Bu, Z., Li, A., Liu, Z., Shi, Y.: Fast and accurate mining the community structure: integrating center locating and membership optimization. *IEEE Trans. Knowl. Data Eng.* **28**(9), 2349–2362 (2016)
14. Liu, C., Chamberlain, B.P.: Speeding up bigclam implementation on snap. arXiv preprint [arXiv:1712.01209](https://arxiv.org/abs/1712.01209) (2017)
15. Nepusz, T., Petróczy, A., Négyessy, L., Bazsó, F.: Fuzzy communities and the concept of bridgeness in complex networks. *Phys. Rev. E* **77**(1), 016107 (2008)
16. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab (1999)
17. Populi, N.: The real-life applications of graph data structures you must know (2018). <https://leapgraph.com/graph-data-structures-applications>
18. Traud, A.L., Kelsic, E.D., Mucha, P.J., Porter, M.A.: Comparing community structure to characteristics in online collegiate social networks. *SIAM Rev.* **53**(3), 526–543 (2011)
19. Traud, A.L., Mucha, P.J., Porter, M.A.: Social structure of Facebook networks. *Phys. A* **391**(16), 4165–4180 (2012)
20. Vanhems, P., Barrat, A., Cattuto, C., Pinton, J.F., Khanafer, N., Régis, C., Kim, B., Comte, B., Voirin, N.: Estimating potential infection transmission routes in hospital wards using wearable proximity sensors. *PLOS ONE* **8**(9) (2013)
21. Xie, J., Kelley, S., Szymanski, B.K.: Overlapping community detection in networks: the state-of-the-art and comparative study. *ACM Comput. Surv. (CSUR)* **45**(4), 1–35 (2013)
22. Yang, J., Leskovec, J.: Overlapping community detection at scale: a nonnegative matrix factorization approach. In: *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, pp. 587–596 (2013)
23. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78**(4), 046110 (2008)
24. Lancichinetti, A., Radicchi, F., Ramasco, J.J., Fortunato, S.: Finding statistically significant communities in networks. *PLOS ONE* **6**(4) (2011)
25. Radicchi, F., Castellano, C., Ceconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. *Natl. Acad. Sci.* **101**, 2658–2663 (2004)
26. Dombi, J., Gera, Z., Vincze, N.: On preferences related to aggregative operators and their transitivity. In: *LINZ*, p. 56 (2006)
27. Dombi, J., Baczyński, M.: General characterization of implication’s distributivity properties: the preference implication. *IEEE Trans. Fuzzy Syst.* **1** (2019)
28. Dombi, J.: Basic concepts for a theory of evaluation: the aggregative operator. *Eur. J. Oper. Res.* **10**(3), 282–293 (1982)
29. Dombi, J., Jónás, T.: Approximations to the normal probability distribution function using operators of continuous-valued logic. *Acta Cybernetica* **23**(3), 829–852 (2018)
30. Csardi, G., Nepusz, T., et al.: The igraph software package for complex network research. *InterJ. Complex Syst.* **1695**(5), 1–9 (2006)
31. Csardi, G.: *igraphdata: A Collection of Network Data Sets for the igraph Package*. r package version 1.0.1 (2015)

32. Rossi, R.A., Ahmed, N.K.: The network data repository with interactive graph analytics and visualization. In: AAAI. <http://networkrepository.com>
33. Callaway, D.S., Newman, M.E., Strogatz, S.H., Watts, D.J.: Network robustness and fragility: percolation on random graphs. *Phys. Rev. Lett.* **85**(25), 5468 (2000)
34. Dombi, J., Dhama, S.: Preference relation and community detection. In: 2019 IEEE 19th International Symposium on Computational Intelligence and Informatics and 7th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Sciences and Robotics (CINTI-MACRo), pp. 33–36 (2019)



Community Detection Algorithm Using Hypergraph Modularity

Bogumił Kamiński¹, Paweł Prałat^{2(✉)}, and François Théberge³

¹ Warsaw School of Economics, Warsaw, Poland
bkamins@sgh.waw.pl

² Ryerson University, Toronto, ON, Canada
pralat@ryerson.ca

³ The Tutte Institute for Mathematics and Computing, Ottawa, ON, Canada
theberge@ieee.org

Abstract. We propose a community detection algorithm for hypergraphs. The main feature of this algorithm is that it can be adjusted to various scenarios depending on how often vertices in one community share hyperedges with vertices from other community.

Keywords: Community detection · Hypergraphs · Modularity

1 Motivation and Our Contribution

An important property of complex networks is their community structure, that is, the organization of vertices in clusters, with many edges joining vertices of the same cluster and comparatively few edges joining vertices of different clusters. In social networks, communities may represent groups by interest (practical application include collaborative tagging), in citation networks they correspond to related papers, and in the web communities are formed by pages on related topics. Being able to identify communities in a network could help us to exploit this network more effectively. Clusters in citation graphs may help to find similar scientific papers, discovering social network users with similar interests is important for targeted advertisement, etc.

Many networks that are currently modelled as graphs would be more accurately modelled as hypergraphs. This includes the collaboration network in which nodes correspond to researchers and hyperedges correspond to papers that consist of nodes associated with researchers that co-authorship a given paper. Unfortunately, the theory and tools are not sufficiently developed to allow most problems, including clustering, to be tackled directly within this context. Indeed, researchers and practitioners often create the 2-section graph of a hypergraph of interest (that is, replace each hyperedge with a clique). After moving to the 2-section graph, one clearly loses some information about hyperedges of size greater than two and so there is a common believe that one can do better by using the knowledge of the original hypergraph.

There are some recent attempts to deal with hypergraphs in the context of clustering. For example, Kumar et al. [6, 7] still reduce the problem to graphs but use the original hypergraphs to iteratively adjust weights to encourage some hyperedges to be included in some cluster but discourage other ones (this process can be viewed as separating signal from noise). Moreover, in [4] a number of extensions of the classic null model for graphs are proposed that can potentially be used by true hypergraph algorithms. Unfortunately, there are many ways such extensions can be done depending on how often vertices in one community share hyperedges with vertices from other communities. This is something that varies between networks at hand and usually depends on the hyperedge sizes. Indeed, hyperedges associated with papers written by mathematicians might be more homogeneous and smaller in comparison with those written by medical doctors who tend to work in large and multidisciplinary teams. Moreover, in general, papers with a large number of co-authors tend to be less homogeneous. A good algorithm should be able to automatically decide which extension should be used.

In this paper, we propose a framework that is able to adjust to various scenarios mentioned above. We do it by generalizing and unifying all extensions of the graph modularity function to hypergraphs, and putting them into one framework in which the contribution from different “slices” is controlled by hyper-parameters that can be tuned for a given scenario (Sect. 2). We propose two prototype algorithms that show the potential of the framework, the so-called proof-of-concept (Sect. 3). In order to test the performance of algorithms in various scenarios, we introduce a synthetic random hypergraph model (Sect. 4) that may be of independent interest. We experiment with our prototypes as well as the two main competitors in this space, namely, the Louvain and Kumar et al. algorithms (Sect. 5). The experiments show that, after tuning hyper-parameters appropriately, the proposed prototypes work very well. Independently, we provide an evidence that such tuning can be done in an unsupervised way. Of course, more work and experiments need to be done before we are able to announce a scalable and properly tuned algorithm but at the end of this paper we reveal a bit more details to that effect (Sect. 6). *Spoiler alert:* the reader who wants to be surprised should avoid that section.

2 Modularity Functions

We start this section by recalling the classic definition of modularity function for graphs (Sect. 2.1). In order to deal with hypergraphs, one may reduce the problem to graphs by considering the corresponding 2-section graph (Sect. 2.2). Alternatively, one may generalize the modularity function to hypergraphs (Sect. 2.3) and then perform algorithms directly on hypergraphs. Such approach should presumably give better results as it preserves more information on the original network in comparison to the corresponding 2-section graphs. In this paper, we generalize the hypergraph modularity function even further that allows us to value various contributions to the modularity function differently (Sect. 2.4).

2.1 Modularity Function for Graphs

Before we define the modularity function, let us introduce some necessary notation and terminology. Let $G = (V, E)$ be a graph where $V = \{v_1, v_2, \dots, v_n\}$ is the set of vertices and E is the set of edges. The edges are multisets of V of cardinality 2 (that is, with loops allowed). Throughout the paper, we will use $n = |V|$ for the number of vertices of G . For a given vertex $v \in V$, $\deg_G(v)$ is the *degree* of v in G (with a loop at v contributing 2 to the degree of v). For $A \subseteq V$, let the *volume* of A be $\text{vol}_G(A) = \sum_{v \in A} \deg_G(v)$; in particular, the volume of the graphs is $\text{vol}_G(V) = \sum_{v \in V} \deg_G(v) = 2|E|$.

The definition of modularity for graphs was first introduced by Newman and Girvan in [11]. Despite some known issues with this function such as the “resolution limit” reported in [3], many popular algorithms for partitioning vertices of large graphs use it [2, 8, 10] and perform very well. The modularity function favours partitions of the vertex set of a graph G in which a large proportion of the edges fall entirely within the parts (often called clusters), but benchmarks it against the expected number of edges one would see in those parts in the corresponding Chung-Lu random graph model which generates graphs with the expected degree sequence following exactly the degree sequence in G .

Formally, for a graph $G = (V, E)$ and a given partition $\mathbf{A} = \{A_1, A_2, \dots, A_k\}$ of V , the *modularity function* is defined as follows:

$$q_G(\mathbf{A}) = \sum_{A_i \in \mathbf{A}} \frac{e_G(A_i)}{|E|} - \sum_{A_i \in \mathbf{A}} \left(\frac{\text{vol}_G(A_i)}{\text{vol}_G(V)} \right)^2, \quad (1)$$

where $e_G(A_i) = |\{\{v_j, v_k\} \in E : v_j, v_k \in A_i\}|$ is the number of edges in the subgraph of G induced by set A_i . The first term in (1), $\sum_{A_i \in \mathbf{A}} e_G(A_i)/|E|$, is called the *edge contribution* and it computes the fraction of edges that fall within one of the parts. The second one, $\sum_{A_i \in \mathbf{A}} (\text{vol}_G(A_i)/\text{vol}_G(V))^2$, is called the *degree tax* and it computes the expected fraction of edges that do the same in the corresponding random graph (the null model). The modularity measures the deviation between the two.

It is easy to see that $q_G(\mathbf{A}) \leq 1$. Also, if $\mathbf{A} = \{V\}$, then $q_G(\mathbf{A}) = 0$, and if $\mathbf{A} = \{\{v_1\}, \{v_2\}, \dots, \{v_n\}\}$, then $q_G(\mathbf{A}) = -\sum (\deg_G(v)/\text{vol}_G(V))^2 < 0$. The maximum *modularity* $q^*(G)$ is defined as the maximum of $q_G(\mathbf{A})$ over all possible partitions \mathbf{A} of V ; that is, $q^*(G) = \max_{\mathbf{A}} q_G(\mathbf{A})$. In order to maximize $q_G(\mathbf{A})$ one wants to find a partition with large edge contribution subject to small degree tax. If $q^*(G)$ approaches 1 (which is the trivial upper bound), we observe a strong community structure; conversely, if $q^*(G)$ is close to zero (which is the trivial lower bound), there is no community structure. The definition in (1) can be generalized to weighted edges by replacing edge counts with sums of edge weights.

2.2 Using Graph Modularity for Hypergraphs

Given a hypergraph $H = (V, E)$, it is common to transform its hyperedges into complete graphs (cliques), the process known as forming the 2-section of

H , graph $H_{[2]}$ on the same vertex set as H . For each hyperedge $e \in E$ with $|e| \geq 2$ and weight $w(e)$, $\binom{|e|}{2}$ edges are formed, each of them with weight of $w(e)/(|e| - 1)$. While there are other natural choices for the weights (such as the original weighting scheme $w(e)/\binom{|e|}{2}$ that preserves the total weight), this choice ensures that the degree distribution of the created graph matches the one of the original hypergraph H [6, 7]. Moreover, let us also mention that it also nicely translates a natural random walk on H into a random walk on the corresponding $H_{[2]}$ [13]. As hyperedges in H usually overlap, this process creates a multigraph. In order for $H_{[2]}$ to be a simple graph, if the same pair of vertices appear in multiple hyperedges, the edge weights are simply added together.

2.3 Modularity Function for Hypergraphs

For the hypergraph $H = (V, E)$, each hyperedge $e \in E$ is a multiset of V of any cardinality $d \in \mathbb{N}$. Multisets in the context of hypergraphs are natural generalization of loops in the context of graphs. Even though H does not always contain multisets, it is convenient to allow them as they may appear in the random hypergraph that will be used to “benchmark” the edge contribution component of the modularity function. It will be convenient to partition the edge set E into $\{E_1, E_2, \dots\}$, where E_d consists of hyperedges of size d . As a result, hypergraph H can be expressed as the disjoint union of d -uniform hypergraphs $H = \bigcup H_d$, where $H_d = (V, E_d)$. As for graphs, $\deg_H(v)$ is the degree of vertex v , that is, the number of hyperedges v is a part of (taking into account the fact that hyperedges are multisets). Finally, the volume of a vertex subset $A \subseteq V$ is $\text{vol}_H(A) = \sum_{v \in A} \deg_H(v)$.

For edges of size greater than 2, several definitions can be used to quantify the edge contribution for a given partition \mathbf{A} of the vertex set. As a result, the choice of hypergraph modularity function is not unique. It depends on how strongly one believes that a hyperedge is an indicator that some of its vertices fall into one community. The fraction of vertices of a given hyperedge that belong to one community is called its *homogeneity* (provided it is more than 50%). In one extreme case, all vertices of a hyperedge have to belong to one of the parts in order to contribute to the modularity function; this is the *strict* variant assuming that only homogeneous hyperedges provide information about underlying community structure. In the other natural extreme variant, the *majority* one, one assumes that edges are not necessarily homogeneous and so a hyperedge contributes to one of the parts if more than 50% of its vertices belong to it; in this case being over 50% is the only information that is considered relevant for community detection. All variants in between guarantee that hyperedges contribute to at most one part. Alternatively, a hyperedge could contribute to the part that corresponds to the largest fraction of vertices. However, this might not uniquely determine the part and it is more natural to classify such edges as “noise” that should not contribute to any part anyway. Once the variant is fixed, one needs to benchmark the corresponding edge contribution using the degree tax computed for the generalization of the Chung-Lu model to hypergraphs proposed in [4].

The framework introduced in [4] is more flexible but, for simplicity, let us concentrate only on the two extreme cases. For $d \in \mathbb{N}$ and $p \in [0, 1]$, let $\text{Bin}(d, p)$ denotes the binomial random variable with parameters d and p . The *majority-based modularity* function for hypergraphs is defined as

$$q_H^m(\mathbf{A}) = \sum_{A_i \in \mathbf{A}} \frac{e_H^m(A_i)}{|E|} - \sum_{d \geq 2} \frac{|E_d|}{|E|} \sum_{A_i \in \mathbf{A}} \mathbb{P} \left(\text{Bin} \left(d, \frac{\text{vol}_H(A_i)}{\text{vol}_H(V)} \right) > \frac{d}{2} \right), \quad (2)$$

and the *strict-based modularity* as

$$\begin{aligned} q_H^s(\mathbf{A}) &= \sum_{A_i \in \mathbf{A}} \frac{e_H^s(A_i)}{|E|} - \sum_{d \geq 2} \frac{|E_d|}{|E|} \sum_{A_i \in \mathbf{A}} \left(\frac{\text{vol}_H(A_i)}{\text{vol}_H(V)} \right)^d \\ &= \sum_{A_i \in \mathbf{A}} \frac{e_H^s(A_i)}{|E|} - \sum_{d \geq 2} \frac{|E_d|}{|E|} \sum_{A_i \in \mathbf{A}} \mathbb{P} \left(\text{Bin} \left(d, \frac{\text{vol}_H(A_i)}{\text{vol}_H(V)} \right) = d \right). \end{aligned} \quad (3)$$

In (2), $e_H^m(A_i)$ counts the number of hyperedges where the majority of vertices belong to part A_i while in (3), $e_H^s(A_i)$ counts the number of edges where all vertices are in part A_i . The goal is the same as for graphs. We search for a partition \mathbf{A} that yields modularity as close as possible to the maximum *modularity* $q^*(H)$ which is defined as the maximum over all possible partitions of the vertex set. We can define weighted versions of the above functions (with weights on hyperedges) the same way as we did for graphs. Finally, note that if H consists only of hyperedges of size 2 (that is, H is a graph), then both (2) and (3) reduce to (1).

2.4 Unification and Generalization

In this section, we unify the definitions of modularity functions and put them into one common framework. This general framework is more flexible and can be tuned and applied to hypergraphs with hyperedges of different homogeneity.

In order to achieve our goal, we “dissect” the modularity function so that each “slice” can be considered independently. For each hyperedge size d , we will independently deal with contribution to the modularity function coming from hyperedges of size d with precisely c members from one of the parts, where $c > d/2$. For example, for $d = 7$ we get 4 slices corresponding to various values of c , namely, $c \in \{4, 5, 6, 7\}$.

Let us first note that (2) can be rewritten as follows:

$$q_H^m(\mathbf{A}) = \sum_{A_i \in \mathbf{A}} \sum_{d \geq 2} \sum_{c=\lfloor d/2 \rfloor + 1}^d \left(\frac{e_H^{d,c}(A_i)}{|E|} - \frac{|E_d|}{|E|} \cdot \mathbb{P} \left(\text{Bin} \left(d, \frac{\text{vol}_H(A_i)}{\text{vol}_H(V)} \right) = c \right) \right),$$

where $e_H^{d,c}(A_i)$ is the number of hyperedges of size d that have exactly c members in A_i . So $q_H^m(\mathbf{A})$ can be viewed as:

$$q_H^m(\mathbf{A}) = \sum_{d \geq 2} \sum_{c=\lfloor d/2 \rfloor + 1}^d q_H^{c,d}(\mathbf{A}),$$

where

$$q_H^{c,d}(\mathbf{A}) = \frac{1}{|E|} \sum_{A_i \in \mathbf{A}} \left(e_H^{d,c}(A_i) - |E_d| \cdot \mathbb{P} \left(\text{Bin} \left(d, \frac{\text{vol}(A_i)}{\text{vol}(V)} \right) = c \right) \right).$$

Similarly, (3) can be viewed as:

$$q_H^s(\mathbf{A}) = \sum_{d \geq 2} q_H^{d,d}(\mathbf{A}).$$

Hence, in the majority-based modularity function q_H^m , each slice is weighted equally whereas for the strict-based definition q_H^s , only the slices with $c = d$ are considered.

In order to unify the definitions, our new modularity function is controlled by *hyper-parameters* $w_{c,d} \in [0, 1]$ ($d \geq 2$, $\lfloor d/2 \rfloor + 1 \leq c \leq d$). For a fixed set of hyper-parameters, we simply define

$$q_H(\mathbf{A}) = \sum_{d \geq 2} \sum_{c=\lfloor d/2 \rfloor + 1}^d w_{c,d} q_H^{c,d}(\mathbf{A}). \quad (4)$$

This definition gives us more flexibility and allows us to value some slices more than others. In our experiments, we restricted ourselves to the following family of hyper-parameters that gave us enough flexibility but is controlled only by 3 variables. (In fact, we will argue later on that one of them, namely ρ_{\max} , can be set to one.) Let $\alpha \in [0, \infty)$, and $\rho_{\min}, \rho_{\max} \in (0.5, 1]$ such that $\rho_{\min} \leq \rho_{\max}$. Then,

$$w_{c,d} = \begin{cases} (c/d)^\alpha & \text{if } \lceil d\rho_{\min} \rceil \leq c \leq \lceil d\rho_{\max} \rceil. \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

Parameters ρ_{\min} and ρ_{\max} are related to the assumption on the minimal and, respectively, maximal ‘‘purity’’ of hyperedges and depends on the level of homogeneity of the network. In particular, ρ_{\max} may be bounded away from one if one expects that ‘‘totally pure’’ (that is, occurring in a single community) hyperedges are unlikely to be observed in practice. Finally, parameter α governs the smooth transition between the relative informativeness between contributing hyperedges of different levels of ‘‘purity’’.

As a result, after adjusting the hyper-parameters accordingly, (4) can be used for the two extreme cases (majority-based and strict-based) and anything in between. Moreover, (4) may well approximate the graph modularity for the corresponding 2-section graph $H_{[2]}$. Indeed, if c vertices of a hyperedge e of size d and weight $w(e)$ fall into one part of the partition \mathbf{A} , then the contribution to the graph modularity is $w(e) \binom{c}{2} / (|e| - 1)$ (in the variant where the degrees are preserved) or $w(e) \binom{c}{2} / \binom{|e|}{2} \approx w(e) (c/|e|)^2$ (if the total weight is preserved). Hence, the hyper-parameters can be adjusted to reflect that. The only difference is that (4) does not allow to include contributions from parts that contain at most $d/2$ vertices which still contributes to the graph modularity of $H_{[2]}$. However, most of the contribution comes from large values of c and so the two corresponding measures are close in practice.

3 Algorithms

In this paper, we experiment with four clustering algorithms that can handle networks represented as hypergraphs. The last two of them are two prototypes of our hybrid and flexible framework under development. More advanced version will be presented in the forthcoming papers but some spoilers are provided in Sect. 6.

3.1 *Louvain*—Graph-Based Algorithm

As discussed in Sect. 2.2, in order to find communities in a hypergraph H , one may reduce the problem to graphs by considering its 2-section (weighted) graph $H_{[2]}$ and then try to find a partition that maximizes the graph modularity function (1) for $H_{[2]}$. One of the mostly used unsupervised algorithms for detecting communities in graphs is the *Louvain* algorithm [1]. It is a hierarchical clustering algorithm that tries to optimize the modularity function (modularity optimization phase), merge communities into single vertices (community aggregation phases), and then it recursively executes the modularity clustering on the condensed graphs until no increase in modularity is possible.

All clustering algorithms are heuristic in nature and only aim to find “good enough” partition without the hope of finding the best one. In particular, in order to be able to search different parts of the solution space, *Louvain* is a randomized algorithm that orders all vertices randomly before the modularity optimization phase takes place. Unfortunately, it means that the algorithm is not stable and outcomes of it may vary significantly between independent runs. In order to solve this issue, the ensemble clustering algorithm for graphs (*ECG*) [12] can be used instead. This algorithm, known to have good stability, is based on the *Louvain* algorithm and the concept of consensus clustering.

3.2 Kumar et al.—Refinement of Graph-Based Algorithm

The following refinement of Kumar et al. [6, 7] generally gives better results than the original *Louvain* algorithm on several synthetic and real-world examples. However, this algorithm is not truly hypergraph-based but should rather be viewed as a refinement of a graph-based approach guided by the original hypergraph. In this algorithm, one first builds a degree-preserving weighted graph G based on the original hypergraph H . Then, the *Louvain* algorithm is applied to G that tries to maximize the graph modularity function (1). After that, hypergraph H is revisited and hyperedges are carefully re-weighted based on their measure of homogeneity between the obtained parts. These steps are repeated until convergence.

3.3 *LS* and *HA*—Our Prototypes

All successful algorithms based on graph modularity optimization (including *Louvain*, *ECG*, and Kumar et al. mentioned above) start the same way. Vertices are initially put into their own clusters, and a basic move is to consider

changing the cluster of a vertex to one of its neighbours' if it increases the modularity function. Unfortunately, trying to apply this strategy to hypergraphs is challenging. Indeed, if one starts from all vertices in their own community, then changing the cluster of only one vertex will likely have no positive effect on the modularity function unless edges of small size are present. For example, it takes several moves for a hyperedge of size $d \geq 4$ to have the majority of its vertices to fall into the same community.

In order to solve this problem, we propose to use the graph modularity function $q_G(\mathbf{A})$ defined in (1) to "lift the process from the ground" but then switch to the hypergraph counterpart $q_H(\mathbf{A})$ defined in (4). There are many ways to achieve it and one of them is mentioned in Sect. 6. For experiments provided in this paper, we consider two prototypes: the first one (*HA*) switches to hypergraphs as soon as possible whereas the second one (*LS*) stays with graphs for much longer. The first algorithm, that we call *HA* (for *hybrid algorithm*), works as follows:

1. Form small, tight clumps by running *ECG* using $q_G(\mathbf{A})$ on the degree-preserving graph G built from H . Prune edges below the threshold value of 70% (number of votes), and keep connected components as initial clumps.
2. Merge clumps (in a random order) if $q_H(\mathbf{A})$ improves. Repeat until no more improvement is possible.
3. Move one vertex at a time (in a random order) to a neighbouring cluster if it improves q_H . Repeat until convergence.

The second algorithm, that we call *LS* (for *last step*) runs Kumar et al. and only does the last step (step 3.) above.

Finally, recall that the hypergraph modularity function $q_H(\mathbf{A})$ is controlled by hyper-parameters $w_{c,d}$ but we restrict ourselves to a family of such parameters guided by parameters α , ρ_{\min} , and ρ_{\max} ; see (5). Hence, we will refer to the above algorithms as $HA(\alpha, \rho_{\min}, \rho_{\max})$ and, respectively, $LS(\alpha, \rho_{\min}, \rho_{\max})$.

4 Synthetic Random Hypergraph Model

In order to test various algorithms in a controlled, synthetic environment, we propose a simple model of a random hypergraph with communities. Such synthetic networks with an engineered *ground truth* are commonly used to evaluate the performance of clustering algorithms. There are many graph models of complex networks, including the well-known and widely used LFR benchmark graph [9] and our own ABCD [5]. On the other hand, very little has been done with hypergraphs. As we aim for a simple model and the degree distribution should not affect our exploratory experiments, we propose a model that is inspired by the classical stochastic block model. Designing more realistic model and performing experiments on it is left for future research.

A random hypergraph \mathcal{H} consists of K communities; the k th community has n_k members so the total number of vertices in \mathcal{H} is equal to $n = \sum_{k=1}^K n_k$. For $2 \leq d \leq M$, m_d is the number of hyperedges of size d ; in particular, M is the

size of a largest hyperedge and $m = \sum_{d \geq 2} m_d$ is the total number of hyperedges. Hyperedges are partitioned into *community* and *noise* hyperedges. The expected proportion of noise edges is $\mu \in [0, 1]$, the parameter that controls the *level of noise*. Each community hyperedge will be assigned to one community. The expected fraction of hyperedges that are assigned to the k th community is p_k ; in particular, $\sum_{k=1}^K p_k = 1$. Community hyperedges that are assigned to the k th community will have majority members from that community. On the other hand, noise hyperedges will be “sprinkled” across the whole hypergraph.

The hyperedges of \mathcal{H} are generated as follows. For each edge size d , we independently generate m_d edges of size d . For each edge e of size d , we first decide if e is a community hyperedge or a noise. It is a noise with probability μ ; otherwise, it is a community hyperedge. If e turns out to be a noise, then we simply choose its d vertices uniformly at random from the set of all sets of vertices of size d , regardless to which community they belong to. On the other hand, if e is a community edge, then we assign it to community k with probability p_k . Then, we fix the homogeneity value τ_e of hyperedge e that is the integer-valued random variable taken uniformly at random from the *homogeneity set* $\{\lceil \tau_{\min} d \rceil, \lceil \tau_{\min} d \rceil + 1, \dots, \lceil \tau_{\max} d \rceil\}$. The homogeneity set depends on parameters τ_{\min} and τ_{\max} of the model that satisfy $0.5 < \tau_{\min} \leq \tau_{\max} \leq 1$, and is assumed to be the same for all edges. Finally, members of e are determined as follows: τ_e vertices are selected uniformly at random from the k th community, and the remaining vertices are selected uniformly at random from vertices outside of this community.

As mentioned above, the proposed model is aimed to be simple but it tries to capture the fact that many real-world networks represented as hypergraphs exhibit various levels of homogeneity or the lack of thereof. Moreover, some networks are noisy with some fraction of hyperedges consisting of vertices from different communities. Such behaviour can be controlled by parameters τ_{\min} , τ_{\max} , and μ . It gives us a tool to test the performance of our algorithms for various scenarios. A good algorithm should be able to adjust to any scenario in an unsupervised way.

5 Experiments

For our experiments we use the synthetic random hypergraph model introduced in Sect. 4. It contains 5 communities, each consisting of 40 vertices: $(n_1, n_2, \dots, n_5) = (40, 40, \dots, 40)$. The distribution of hyperedge sizes is as follows: $(m_1, m_2, \dots, m_{11}) = (30, 30, 30, 30, 30, 30, 30, 20, 20, 20)$. The expected fraction of edges that belong to a given cluster is equal to 0.2: $(p_1, p_2, \dots, p_5) = (0.2, 0.2, \dots, 0.2)$. The lower bound for the homogeneity interval is fixed to be $\tau_{\min} = 0.65$. We performed experiments on four hypergraphs with the remaining two parameters fixed to: a) $(\mu, \tau_{\max}) = (0, 0.65)$, b) $(\mu, \tau_{\max}) = (0, 0.8)$, c) $(\mu, \tau_{\max}) = (0, 1)$, d) $(\mu, \tau_{\max}) = (0.1, 0.80)$. All of them lead to the same conclusion so we present figures only for hypergraph \mathcal{H} that is obtained with parameters d).

We test the two known algorithms, Louvain and Kumar et al., as well as our two prototypes, LS and HA . For each prototype, we test three different sets of hyper-parameters. In the first variant, we include contribution to the hypergraph modularity function that comes from all slices, that is, we fix $\rho_{\min} = 0.5^+ = 0.5 + \epsilon$ (for some very small $\epsilon > 0$ so that all “slices” of the modularity function are included) and $\rho_{\max} = 1$. For simplicity, we fix $\alpha = 1$. For convenient notation, let $LS = LS(1, 0.5^+, 1)$ and $HA = HA(1, 0.5^+, 1)$. For the second variant, we use the knowledge about the hypergraph (*ground truth*) and concentrate only on slices that are above the lower bound for the homogeneity set, that is, we fix $\rho_{\min} = \tau_{\min} = 0.65$ but keep $\rho_{\max} = 1$. Let $LS+ = LS(1, 0.65, 1)$ and $HA+ = HA(1, 0.65, 1)$. Finally, we use the complete knowledge about the generative process of our synthetic hypergraph and fix $\rho_{\min} = \tau_{\min} = 0.65$ and $\rho_{\max} = \tau_{\max} = 0.80$. The corresponding algorithms are denoted by $LS++ = LS(1, 0.65, 0.85)$ and $HA++ = HA(1, 0.65, 0.85)$.

In the first experiment, we run each algorithm on \mathcal{H} and measure its performance using the *Adjusted Mutual Information* (AMI). AMI is the information theory measure that allows us to quantify the similarity between two partitions of the same set of nodes, the partition returned by the algorithm and the *ground truth*. Since all algorithms involved are randomized, we run them 100 times and present a box-plot of the corresponding AMIs in Fig. 1(a). We see that LS and HA give comparable results as the original Louvain and Kumar et al. is consistently better. On the other hand, when our prototypes are provided with a knowledge about the homogeneity of \mathcal{H} , they perform very well, better than Kumar et al. There is a less difference between $+$ and $++$ variants of the two prototypes. This is a good and desired feature as “pure” hyperedges should not generally be penalized unless there is some known external hard constraint that prevents hyperedges to be homogeneous. On the other hand, if large hyperedges are non-homogeneous, then the quality of $+$ and $++$ should be similar as these “slices” barely contribute to the modularity function anyway. Note that this observation does not apply to small hyperedges; in the extreme situation when dealing with graphs with hyperedges of size 2, any choice of $\rho_{\max} \geq \rho_{\min}$ leads to exactly the same results.

The previous experiment shows that knowing some global statistics (namely, how homogeneous the network is) significantly increases the performance of our prototypes. However, typically such information is not available and the algorithm has to learn such global statistics in an unsupervised way. In our second experiment, we test if this is possible. We take a partition returned by Kumar et al. and investigate all hyperedges of \mathcal{H} . For each hyperedge e we check if at least $\tau \geq 0.5$ fraction of its vertices belong to some community. We compare it with the corresponding homogeneity value based on the *ground truth*. The two distributions are presented in Fig. 1(c) and are almost indistinguishable. This suggests that learning the right value of ρ_{\min} should be possible in practice.

Finally, we tested the performance of our prototypes for various choices of parameter α . $+$ and $++$ variants turn out to be not too sensitive whereas LS and HA increase their performance as α increases—Figure 1(b). It is perhaps not

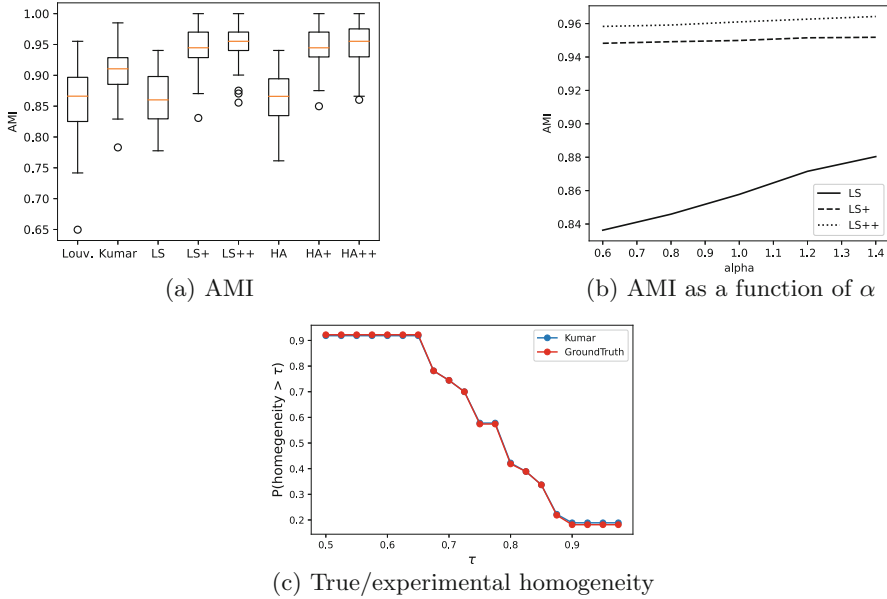


Fig. 1. Experiments on hypergraph \mathcal{H} : $\mu = 0.1$, $\tau_{\min} = 0.65$, $\tau_{\max} = 0.8$.

too surprising as increasing α puts more weight to more homogeneous hyper-edges which has similar effect to tuning parameter ρ_{\min} . More comprehensive experiments are to be performed.

6 Conclusions and Future Directions

In this paper, we propose two prototype algorithms and do some simple experiments that show their potential (of course, we experimented much more and most experiments are encouraging). Clearly more work needs to be done. For example, we showed that our prototypes work very well but only once proper tuning of the hyper-parameters is done. Initial experiments show that such tuning can be done in an unsupervised way but details need to be fixed. One important thing that we keep in mind is a potential risk of a solution to be overfitted.

We proposed two ways to solve a problem with initial phase of any algorithm based on the hypergraph modularity function, our two prototypes. Another option is to embed vertices of the 2-section graph in a geometric space such that nearby vertices are more likely to share an edge than those far from each other. Then, for example, the classical *k-means* algorithm with some large value of k may be used to find the initial partition and then one can switch to the hypergraph based algorithms optimizing the hypergraph modularity function.

Hyperedge re-weighting scheme proposed by Kumar et al. seems to work very well. This is an independent component that can be easily incorporated within our framework. We aim for a flexible framework that can mimic Louvain, ECG,

Kumar et al., and anything in between, but is additionally enhanced by the opportunities provided by the hypergraph modularity function.

The algorithm has to be scalable so that we may run it on large hypergraphs. The updates of the hypergraph modularity function can be done fast but it requires a proper design/usage of dedicated data structures and algorithms. We currently implement such a code in the Julia language.

Finally, on top of experimenting on large synthetic hypergraphs we plan to perform experiments on real-world networks represented as hypergraphs.

References

1. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech. Theor. Exp.* **10**, P10008 (2008)
2. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. *Phys. Rev. E* **70**, 066111 (2004)
3. Fortunato, S., Barthelemy, M.: Resolution limit in community detection. *Proc. Natl. Acad. Sci. U.S.A.* **104**, 36–41 (2007)
4. Kamiński, B., Poulin, V., Prałat, P., Szufel, P., Théberge, F.: Clustering via hypergraph modularity. *PLOS ONE* **14**(11), e0224307 (2019)
5. Kamiński, B., Prałat, P., Théberge, F.: Artificial Benchmark for Community Detection (ABCD)—Fast Random Graph Model with Community Structure, [arXiv:2002.00843](https://arxiv.org/abs/2002.00843)
6. Kumar, T., Vaidyanathan, S., Ananthapadmanabhan, H., Parthasarathy, S., Ravindran, B.: A new measure of modularity in hypergraphs: theoretical insights and implications for effective clustering. In: *International Conference on Complex Networks and Their Applications, Complex Networks 2019*, pp. 286–297. Springer, Cham (2019)
7. Kumar, T., Vaidyanathan, S., Ananthapadmanabhan, H., Parthasarathy, S., Ravindran, B.: Hypergraph clustering by iteratively reweighted modularity maximization. *Appl. Netw. Sci* **5** (2020)
8. Lancichinetti, A., Fortunato, S.: Limits of modularity maximization in community detection. *Phys. Rev. E* **84**, 066122 (2011)
9. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78** (2008)
10. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. *Phys. Rev. E* **69**, 066133 (2004)
11. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. *Phys. Rev. E* **69**, 026–113 (2004)
12. Poulin, V., Théberge, F.: Ensemble clustering for graphs. In: Aiello, L., Cherifi, C., Cherifi, H., Lambiotte, R., Lió, P., Rocha, L. (eds.) *Complex Networks and their Applications VII, COMPLEX NETWORKS 2018*. Studies in Computational Intelligence, vol. 812. Springer, Cham (2018)
13. Théberge, F.: Summer School on Data Science Tools and Techniques in Modelling Complex Networks. <https://github.com/ftheberge/ComplexNetworks2019/>



Towards Causal Explanations of Community Detection in Networks

Georgia Baltso¹(✉), Anastasios Gounaris¹, Apostolos N. Papadopoulos¹,
and Konstantinos Tsichlas²

¹ School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
{georgipm,gounaria,papadopo}@csd.auth.gr

² Department of Computer Engineering and Informatics, University of Patras,
Patras, Greece
ktsichlas@ceid.upatras.gr

Abstract. Community detection is a significant research problem in Network Science since it identifies groups of nodes that may have certain functional importance - termed communities. Our goal is to further study this problem from a different perspective related to the questions of the cause of belongingness to a community. To this end, we apply the framework of causality and responsibility developed by Halpern and Pearl [11]. We provide an algorithm-semi-agnostic framework for computing causes and responsibility of belongingness to a community. To the best of the authors' knowledge, this is the first work that examines causality in community detection. Furthermore, the proposed framework is easily adaptable to be also used in other network processing operations apart from community detection.

Keywords: Causality · Network analysis · Community detection · Algorithms

1 Introduction

Imagine someone participating in a social network. Due to an analytics engine that the social network offers for its users, she finds out that she is unintentionally part of a community and asks what are the reasons for her belongingness to this community. She would also wish to become a member of another community - always in the context of the community detection algorithm offered by the analytics engine - and asks what new relations she would have to set up in order to become a member. Note that in this example, one's membership in a community is not explicit but implicit through the social network analytics engine, which affects many aspects of the user's belongingness to the social network (e.g., recommendation of new friends, selection of ads to show, etc.) and thus it is of high importance to the user. In particular, we ask:

1. *What causes the fact that a node u belongs to a community C ? Which are the edges that are responsible for $u \in C$? Can we rank these edges based on the degree of their responsibility for $u \in C$?*

2. *What causes the fact that a node u does not belong to a community C ? Which are the new edges that would allow u to become a member of C ?*

Networks are used to represent data in almost any field, such as transportation systems [7], biological systems [22], and social groups [20], just to name a few. In such networks, certain groups of nodes with particular importance arise, which form the so called communities. The dominant definition of community is a group of nodes that are more densely connected internally than externally [25]. In real-world networks, communities are of major importance since they are related to functional units [3,24]. Communities have also topological properties that are different from those of the network as a whole.

In this work, we formulate such questions related to community detection by using the *structural-model approach* introduced by Halpern and Pearl [11,12]. In particular, we focus on the community detection problem and we define different sets of causes with different degrees of importance with respect to the question at hand. This importance is captured by a measure of *responsibility* [4] for each cause, thus allowing for a ranking of the causes. Moreover, this structural-model approach has the side-effect that other communities may change as a result of a question for a particular node. We introduce a measure for these changes to quantify how the interventions implied by the cause alter the community structure of the network. To the best of our knowledge, we are the first to look at the community detection problem on networks through the lens of causal explanations. In fact, it seems that this is the first time that such a viewpoint is adopted with respect to general network analysis problems.

Related Work. Community detection in general has been a very active field during the last years. There is a plethora of algorithms aiming at finding the best quality communities in networks, based on different evaluation metrics. Those works include both disjoint or overlapping community detection algorithms. For some detailed surveys on the field we refer to [8,14]. However, there is no work on combining causal explanations and community detection. The structural-model approach introduced by Halpern and Pearl [11,12] has been applied mainly to database queries. Meliou et al. [18] transferred these notions to databases. This approach is related to data provenance, lineages and view updates (e.g., deletion propagation) [19]. Inspired by this approach others have applied this structural-model approach to reverse-skyline queries [10], to probabilistic nearest neighbor queries [16], and so on. One more network-related problem where this model has been applied concerns the ranking of propagation history in information diffusion in social networks [27].

Contributions and Roadmap. This work focuses on the application of causality in the community detection problem. Examining causality in community detection in networks is novel in its own right. We suggest a general framework that can be used to find causal relations about the belongingness of nodes to communities. An interesting aspect is that the proposed framework is easily adaptable to other network processing operations apart from community detection. Apart from transferring the concepts in [11,12], we also introduce the con-

cept of discrepancy, as a measure of network changes which occur after specific actions.

The rest of the work is organized as follows. In Sect. 2 we discuss how the causal model of Halpern and Pearl applies to community detection while in Sect. 3 we provide a general framework that can compute causal explanations and related metrics. In Sect. 4 we discuss additional issues and future extensions of the proposed framework.

2 The Causal Model for Community Detection

Here we study the proposed causal model and introduce some fundamental concepts.

2.1 Preliminaries

Initially, we restrict our setting to a simple undirected, unweighted network $G = (V, E)$, which is composed of a node set $V = \{1, \dots, n\}$ with $n = |V|$ nodes and an edge set $E \subseteq |V| \cdot |V - 1|$ with $m = |E|$ edges; we discuss extensions to more generic graphs in Sect. 4. Let $G[S]$ represent the induced sub-graph of the node set S , $S \subseteq V$. The adjacent nodes of a node u , i.e., the nodes connected to u with an edge, are its neighbours: $N(u) = \{u|\{u, v\} \in E\}$. The degree of u is $deg(u) := |N(u)|$, i.e., the number of u 's neighbours.

Modularity [21] is a widely used objective function to measure the quality of a network's decomposition into communities and is defined as:

$$Q = \sum_{C=1}^j \left[\frac{m_C}{m} - \left(\frac{deg^C}{2m} \right)^2 \right]$$

where j is the number of communities in the network, m_C is the number of intra-community edges of C and deg^C is the sum of degrees of all nodes in C .

2.2 The Proposed Approach

The definition of causality is based on the work of Halpern and Pearl [11]. Based on their definition of actual causality we identify and analyze three main concepts within our context: *i*) endogenous and exogenous pairs of nodes, *ii*) contingency sets and *iii*) responsibility.

In general, the fact that a node, henceforth termed as the *query node*, belongs to a community is mainly determined by its incident edges. However, the non-incident edges affect the composition of query's node community and as a result they can affect indirectly the node's belongingness to it. Similar arguments hold for the non-belongingness of a node to a community.

In a nutshell, we try to identify the existing edges that result in the user's v belongingness to a community. Similarly, we try to pinpoint the non-existent incident edges of v that could put v in a new community. To this end, all possible pairs of nodes in $|V| \cdot |V - 1|$ in the network can be partitioned into *exogenous* and

endogenous ones.¹ Exogenous pairs of nodes $E_x \subseteq |V| \cdot |V - 1|$ are not considered to have a causal effect on the (non-)belongingness of node v to a community and endogenous $E_e \subseteq |V| \cdot |V - 1|$ are the pairs of nodes that can in principle infer such causal implications. Note that $E_x \cup E_e = |V| \cdot |V - 1|$ and $E_x \cap E_e = \emptyset$.

To check if an edge e is a cause for the (non-)belongingness of a node v to a particular community C , we have to find a set of endogenous pairs of nodes whose edge removal/addition will allow e to immediately affect the belongingness of v to the community C . These sets are called *contingencies*. Note that the contingency set does not alone change the community of v but it is required in order to unlock the causal effect of edge e on the belongingness of v . The contingency set must be minimal, in a manner that removing any edge from it, will dampen the causal effect of e to the belongingness of v to its community, so, no redundancy is allowed.

In a sense, all incident edges (and possibly additional ones) of v affect its belongingness to the community (either in positive or negative manner). Thus, we need a ranking function that will allow us to reason about the most important causes for v (non-)participating in the community. *Responsibility* [4] measures the degree of causality of an edge e for a node v as a function of the smallest contingency set.

In the following, we provide a definition of causality tailored to the problem of why a node belongs to a particular community.

Definition 1. Let $e \in E_e$ be the edge connecting an endogenous pair of nodes and let v belong to community C .

- e is called a *counterfactual cause* for $v \in C$ if for the network $G = (V, E)$ it holds that $v \in C$ while for $G' = (V, E - \{e\})$ it holds that $v \notin C$.
- e is called an *actual cause* for $v \in C$ if there exists a set of edges $\Gamma \subseteq E_e$ called a *contingency* for e such that e is a counterfactual cause for $v \in C$ in the network $G' = (V, E - \Gamma)$.

Next, we provide a definition of causality tailored to the problem of explaining why node v is not a member of a community C' .

Definition 2. Let $e \in E_e/E$ a non-existent edge and C' a community that does not contain v .

- e is called a *counterfactual cause* for $v \notin C'$ if for the network $G = (V, E)$ it holds that $v \notin C'$ while for $G' = (V, E + \{e\})$ it holds that $v \in C'$.
- e is called an *actual cause* for $v \notin C'$ if there exists a set of edges $\Gamma \subseteq E_e$ such that $\Gamma \cap E = \emptyset$ called a *contingency* for e such that e is a counterfactual cause for $v \notin C'$ in the network $G' = (V, E + \Gamma)$.

¹ The endogenous and exogenous sets differ for different nodes. Also, the endogenous set typically comprises the incident edges connecting this node to its neighbors. Finally, allowing self-loops is not an issue, since if they are irrelevant to the setting, we can simply make them exogenous.

Finally, based on [4] we provide a measure of the degree of causality, thus providing a ranking function for the various causes.

Definition 3. Let v be the query node with respect to a community C in the network $G = (V, E)$ and let the set of edges e be a cause. The responsibility of e for v participating or not in C is:

$$\rho_e = \frac{1}{1 + \min_{\Gamma} |\Gamma|}$$

for all contingency sets Γ for e , where $|\Gamma|$ is the size of the set Γ .

The domain of ρ_e is in $(0, 1]$. If the contingency set is \emptyset , then the responsibility is 1, otherwise, the larger the contingency set the less the responsibility. In this way, we capture the degree of interventions needed (the set Γ) to uncover the causal implication of e on v with respect to community C .

At this point we need to discuss a distinguishing feature in the introduction of causality in community detection. The counterfactual interventions suggested by the contingency set and the cause may as well change other communities. This may seem as an undesirable side-effect of our definition that we may choose to ignore, as the question of the causes for the belongingness or non-belongingness of v to community C is related to v alone, and so potential changes to other communities are of no interest to v . Since these causes are counterfactual, in fact no change happens if they are simply used for the purpose of briefing v .

However, if we consider v to be an agent whose purpose is to find out what actions should be taken in order to achieve her removal from C (in the case she asks of the causes of her belongingness to C) or her addition to C' (in the case she asks of the causes of her non-belongingness to C') then this side-effect becomes important. In this case, the causes and their corresponding contingency sets can be considered as a suggested set of actions so that v achieves her goal. Apparently, the endogenous set must be defined so that v can alter the corresponding edges. Going into more depth, one will confront various issues like the identity problem that comes up in community detection in temporal networks [23]; that is, after the intervention, what happens if C has changed so much that cannot be considered as C anymore? We avoid such issues by introducing a measure of such changes, called *discrepancy*.

Definition 4. Let v be a query node with respect to community C in the network $G = (V, E)$. Let V_c be the set of nodes, excluding v , that change community as a result of the intervention implied by the cause e and its corresponding contingency set Γ . Then, the discrepancy $\gamma(v, e, \Gamma)$ of v with respect to e and Γ is defined as:

$$\gamma(v, e, \Gamma) = \frac{|V_c|}{|V| - 1}$$

The domain of the discrepancy is $[0, 1]$. If it is zero then no node changes and thus $|V_c| = 0$. If all nodes change then $|V_c| = |V| - 1$ and thus discrepancy is equal to 1.

Finally, we make three assumptions for efficiency and effectiveness purposes. These assumptions mostly affect the algorithmic aspects discussed in the next section, but are given here as part of the core proposal. The first assumption that was already implied in the discussion of endogenous pairs of nodes, concerns the edges that constitute causes of the (non-)belongingness of node v to a community C . This is the *Locality Assumption*.

Assumption 1. *The more distant two nodes the less they influence each other.*

This assumption allows us to focus on possible causes around node v and in the involved communities. Pairs of nodes whose corresponding edges are considered far from v and do not belong to the involved communities are not considered as endogenous. In community detection in unweighted networks, this assumption is part of its very definition, since the belongingness of node v to a community is guided mainly by its incident edges. Such an assumption is widely used in network analysis [2, 5, 6], e.g., in social networks is known as the *Friedkin's postulate* [9].

We also make an assumption concerning the size of communities, called henceforth the *Size Assumption*. This assumption allows us to bound the number of endogenous pairs of nodes introduced by the involved communities.

Assumption 2. *Communities are polynomially smaller than the size of network.*

Usually, communities tend to be smaller than the size of the network. As discussed in [8], after systematic analysis by the authors of [15], communities in many large networks, including traditional and online social networks, technological, information networks and web graphs, are fairly small in size. It is also believed that the communities in biological networks are relatively small i.e., 3–150 nodes [26, 28]. We capture this phenomenon by assuming that the size of communities is $O(n^\epsilon)$ for some small constant $\epsilon < 1$.

Finally, for efficiency reasons, we assume that both the number of causes and the size of the Γ set are small. We call this assumption the *Bound Assumption*.

Assumption 3. *The number of causes and the size of contingency sets are bounded by a small constant.*

This assumption is important because it limits the available options for causes and the contingency set. If the number of causes was large, then the information on the causal relations would be minuscule. Besides, if the size of the contingency sets were large, that would lead to a very low value of responsibility, meaning that the effect of the actual cause is minuscule.

Example. We discuss a simple example to provide a foothold to move to the framework description in the next section. In Fig. 1 the friendships between members of the Zachary Karate club [29] are shown. The Louvain method [1] has been used to partition the network into 4 communities. Note that in reality the *Green*

and *Orange* communities are the one group after the division while the *Blue* and *Purple* communities correspond to the other group. The modularity Q of this network decomposition is 0.417.

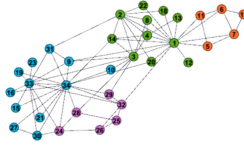


Fig. 1. The Zachary karate club network. There are 4 different communities denoted by different colors: *Green*, *Orange*, *Blue* and *Purple*.

Let us first look at node 10. What is the cause for $10 \in \textit{Blue}$? Removing edge $(10, 34)$ apparently leads to 10 not belonging anymore in *Blue* but in *Green* and thus edge $(10, 34)$ is a counterfactual cause. This is the case with modularity $Q = 0.427$ and no other node changes community, which means that discrepancy $\gamma = 0$ while responsibility $\rho = 1$, since the contingency set is empty. The Locality Assumption 1 was used since the endogenous pairs of nodes were assumed to be only the neighbours of node 10. In case we extend the set of endogenous pairs to contain the neighbours of 34, we could weaken node 34, by choosing some of its incident edges (with the exception of edge $(34, 10)$), thus indirectly making node 10 belong to *Green*. However, this is not a cause for $10 \in \textit{Blue}$ but a by-product of 34 being a hub node of *Blue*. Of course, by transitivity,² the fact that 34 is a hub of *Blue* causes $10 \in \textit{Blue}$ through edge $(34, 10)$, but we prefer to look straightforwardly at the direct cause expressed by this edge.

Why does node $32 \notin \textit{Blue}$? As seen in Fig. 1, node 32 is quite central in *Purple* community. We found out that the edge $(31, 32)$ is a cause for $32 \notin \textit{Blue}$ with contingency $\Gamma = \{(19, 32)\}$ meaning that its responsibility for $32 \notin \textit{Blue}$ is $1/2$. Note that since the network is undirected the same can be said for edge $(19, 32)$ as a cause with contingency $\Gamma = \{(31, 32)\}$ with $\rho = 1/2$. In this case $Q = 0.404$ while node 29 is also put in *Blue* community, and thus $\gamma = 1/34$.

Finally, lets look at node 9. Why does $9 \notin \textit{Green}$? Iterating over all nodes in *Green* as causes we get the results in Table 1. The results are expected since $N(9) = \{1, 3, 31, 33, 34\}$, which are the most central nodes w.r.t. degree in their communities. We expected that $(9, 2)$ would be a counterfactual cause for $9 \notin \textit{Green}$ but this is not the case.

What if we extend the definition of contingency and allow for deletions of edges of 9 to nodes within its current community so that its belongingness to *Blue* community is weakened? Then, in this case the edge $(9, 34)$ is a cause with $\Gamma = \{(9, 31)\}$ since their removal moves 9 to *Green* with $\gamma = 0$ and $Q = 0.423$.

² Transitivity does not hold in general w.r.t. causation [11].

Table 1. Causes for node $9 \notin Green$. ρ corresponds to responsibility, Q to modularity and γ to discrepancy. Not all nodes of *Green* are shown since they have exactly the same behavior.

Cause	Γ	ρ	Q	γ
(9, 12)	$\{(9, 2)\}$	$\frac{1}{2}$	0.402	0
(9, 20)	$\{(9, 2)\}$	$\frac{1}{2}$	0.402	0
(9, 14)	$\{(9, 2)\}$	$\frac{1}{2}$	0.402	0
(9, 4)	$\{(9, 2)\}$	$\frac{1}{2}$	0.402	0
(9, 8)	$\{(9, 2)\}$	$\frac{1}{2}$	0.402	0
(9, 2)	$\{(9, 8)\}$	$\frac{1}{2}$	0.402	0

3 Algorithmic Aspects

In this section we describe algorithmic aspects that allow us to answer why (Definition 1) and why-not (Definition 2) queries for community detection. We first provide a general framework that is oblivious to the community detection algorithm being used. Then, within this framework and for reasons of efficiency, we specialize by focusing on modularity-based algorithms.

3.1 The General Framework

We begin by describing a trivial algorithm-agnostic framework. In fact, this framework is so general that can be used as a first step for introducing causality in different network settings as argued in Sect. 4. Assume an algorithm \mathcal{A} that divides a given network $G = (V, E)$ into a set of communities \mathcal{C} . We pose the question “why a node $v \in V$ belongs in community $C \in \mathcal{C}$ ” (henceforth *why question*). For the *why-not question* the framework works in the same way.

Following Definition 1, we need to identify edges within E_e that are causes and discover their respective contingency sets Γ as well as the changes implied by them in the community structure in order to compute the responsibility ρ and the discrepancy γ . To accomplish this, we first iterate over all subsets c of E_e to choose possible causes e in increasing size (starting from singletons) and then we iterate on all subsets of E_e/e to compute Γ . We maintain the top- k causes with highest ρ . If we are interested on γ as well, we could use either a weighted mean or maintain the top- k dominating causes with respect to both metrics. A very crude upper bound for the method is $O(2^{2y})$ iterations of the algorithm \mathcal{A} , where $y = |E_e|$ is the number of endogenous pairs of nodes.

Apparently, the time complexity of this framework is prohibitive. To speed the algorithm up, we can use the Locality Assumption. In this sense, we can define the endogenous pairs of nodes to be all corresponding edges at a small distance from v . For example, if we include in the endogenous set the neighbours of v , then the number of iterations is $O(2^{2deg(v)})$, which is considerably smaller especially for sparse networks that are usually seen in practice. However, even in this case the number of iterations is quite large. We could further

reduce the complexity by having some information about the inner workings of the algorithm \mathcal{A} . In the following, we assume such an approach by looking at an algorithm that optimizes modularity. In addition, for the *why question* we consider as endogenous pairs of nodes all the neighbours of the query node v .

3.2 Working with Modularity-Based Methods

Firstly, we apply a modularity based community detection algorithm in the given network G , such as the Louvain method, which maximizes modularity. We refer the reader to [1] for more details. G is now partitioned into communities.

We focus on the *why question*. Subsequently we have to decide which edges will be examined as possible causes. Therefore, we use a combination of two metrics: *embeddedness* (ξ_v) and *degree* ($deg(v)$) of the query node v . The embeddedness ξ_v of v in community C , is defined as the ratio between the number of edges connecting v to nodes of C , and the degree of v [8], i.e., $\xi_v = deg^C(v)/deg(v)$. The higher the value of ξ_v , the stronger the belongingness of v to C .

However, this metric alone cannot be used in our case because it is misleading. Let's look at the example of Fig. 1. The embeddedness of node 2 in the Green community is approximately equal to 0.89. On the other hand, the embeddedness of node 12 in the same community is equal to 1. However, node 12 has only one edge and it is rational for this edge to be incident to a node of the same community. Thus, we can combine embeddedness with the degree of the query node resulting in a metric M as follows: $M_v = \frac{\xi_v \cdot deg^C(v)}{\max(deg^C)}$, where $\max(deg^C)$ is the maximum node degree inside community C . As it can be understood, metric M is defined as above in order to reward edges that participate more actively in their community. It is also a simple metric that can be easily implemented. Note that instead of M , we can use any other metric. Consequently, we rank the edges by their M values in decreasing order, and consider as cause(s) the first x edge(s) of this ranking. The constant x is defined by the maximum number of causes as it has been assumed by the Bound Assumption. Then, we compute the corresponding ρ and γ values.

Now the structure of G has changed due to the interventions Δ , implied by the above causes and their Γ . Thus, we must apply again community detection in the new network G' , which is G after the integration of Δ . As it may be inferred, the changes of G are not so radical and are observed to be around specific parts of G . For this reason, we can apply the Louvain method only to a part of G whose community affiliation might change due to the Δ . There are some incremental community detection approaches such as [13, 30] that can be implemented along with either Louvain or any other modularity based community detection method.

4 Additional Issues and Extensions

In this section, we discuss various extensions to the framework discussed above.

Weighted Networks. The proposed approach can be extended to weighted networks as well. An undirected, weighted network $G = (V, E, w)$ is composed of

a node set $V = \{1, \dots, n\}$ with $n = |V|$ nodes and an edge set $E \subseteq V \times V$ with $m = |E|$ undirected edges and edge weights $w = E \rightarrow \mathbb{R}_{>0}$. Definitions 1 and 2 are straightforward to apply. The weighted responsibility is a simple extension of the unweighted case as we define $\Gamma = \sum_{e \in \Gamma} w(e)$.

In weighted networks, where the weight corresponds to how strongly two nodes are connected, the Locality Assumption implies that paths with large total additive weight are preferred over paths of lower weight. This is because it has been assumed that weights resemble similarity and not distance, in which case one has to consider the inverse of weights. The major difference is that in the unweighted case the choice for an edge is binary (remove/add). In the weighted case, the choice is not binary since the algorithms to identify causes must also be able to increase/decrease weights; e.g., in a social network these changes in weights may correspond to the strengthening/weakening of a friendship. This requires a strategy to handle these weights and affects the discrepancy measure.

Uncertain Networks. Our approach is naturally extended to the case of uncertain networks. An uncertain network $G = (V, E, P)$ is defined over a set of nodes V , a set of edges E between pairs of nodes and a probability distribution P over the edges E . Definitions 1 and 2 as well as ρ and γ can be straightforwardly generalized to uncertain networks, e.g., we can simply change the probability of existence of an edge and increase it or decrease it in order to prove actual causes. The approach will be very similar to the case of the weighted networks with additional restrictions related to handling probabilities.

Extending the Definition of Contingency. The contingency sets may be different considering the accepted actions we can do i.e. addition/removal of an edge, weight changes, etc. Note that in Definition 1, Γ contains edges to be removed from the network. Γ could also include, if necessary, edges to be added. Adding edges in this case strengthens the node's belongingness to other communities, thus moving it further away from community C . Similarly, in Definition 2, Γ contains edges to be added to the network. Γ could also include, if necessary, edges to be removed that could indirectly lead v to belong to another community. Besides, if the network is weighted, the contingency set may be altered if we consider the changes on the edges' weights. Although these extended definitions would provide more options, efficiency would be aggravated.

The Endogenous Pairs of Nodes. In general, for the *why question*, we can consider as endogenous any pair of nodes whose corresponding edges are incident to the query node. Furthermore, we can expand the former by adding edges that belong to the same community as the query node. For the *why-not question*, we can add to the endogenous set, pairs of nodes that belong to neighbouring communities of the query node's community. The choice of the endogenous pairs of nodes is of critical importance for the efficiency and depends heavily on the definitions of the actual cause and the contingency set as exemplified in our previous point. In addition, this choice affects how much freedom the algorithm will have in order to identify causes and especially surprising causes. A surprising cause would be a distant edge to node v whose removal would lead v to change

its community according to algorithm \mathcal{A} . This trade-off needs to be handled carefully, as the more wider the endogenous sets, the more the availability of possible causes but at the same time the more processing time will be needed.

Beyond Community Detection. The general but inefficient framework proposed in Sect. 3, can be readily extended to other network-related processing problems as well. For example, asking why a node belongs to the k -core of the network [17] can be tackled by this framework given that the appropriate changes have been made to Definitions 1–4. In the following, we show how Definition 1 would change in this case.

Definition 5. Let $e \in E_e$ be an endogenous pair of nodes and let v belong to the k -core.

- e is called a counterfactual cause for v in the k -core if for the network $G = (V, E)$ it holds that v belongs to the k -core while for $G' = (V, E - \{e\})$ it holds that v does not belong to the k -core.
- e is called an actual cause for v belonging in the k -core if there exists a set of edges $\Gamma \subseteq E_e$ called a contingency for e such that e is a counterfactual cause for v belonging in the k -core in the network $G' = (V, E - \Gamma)$.

In this case, E_e could contain all edges in the k -core of the network since these are the possible causes for node v being in the k -core. Similarly, one can introduce causality in the minimum cut problem in a weighted network (why does edge e belong to the cut?). Efficiency issues must be handled in an ad-hoc manner based on the problem at hand.

In the present work, we have introduced the concept of causal explanations in community formation and we have proposed a framework for identifying actual causes. In the future, we will focus on efficient algorithmic techniques as well as on extensive experimental evaluation for different types of networks (e.g., directed, weighted) and different problems (e.g., overlapping communities, k -core decomposition).



Acknowledgements. Georgia Baltsou is co-financed by Greece and the European Union (European Social Fund-ESF) through the Operational Programme “Human Resources Development, Education and Lifelong Learning” in the context of the project “Strengthening Human Resources Research Potential via Doctorate Research - 2nd Cycle” (MIS-5000432), implemented by the State Scholarships Foundation (IKY).

References

1. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech. Theor. Exp.* **2008**(10), P10008 (2008)

2. Carmi, E., Oestreicher-Singer, G., Sundararajan, A.: Is Oprah contagious? Identifying demand spillovers in online networks. Identifying Demand Spillovers in Online Networks (August 2012). NET Inst. Working Paper No. 10-18 (2012)
3. Chen, S., Wang, Z.Z., Tang, L., Tang, Y.N., Gao, Y.Y., Li, H.J., Xiang, J., Zhang, Y.: Global vs local modularity for network community detection. *PLoS ONE* **13**(10), e0205284 (2018)
4. Chockler, H., Halpern, J.Y.: Responsibility and blame: a structural-model approach. *J. Artif. Intell. Res.* **22**(1), 93–115 (2004)
5. De Meo, P., Ferrara, E., Fiumara, G., Provetti, A.: Mixing local and global information for community detection in large networks. *J. Comput. Syst. Sci.* **80**(1), 72–87 (2014)
6. De Meo, P., Ferrara, E., Fiumara, G., Ricciardello, A.: A novel measure of edge centrality in social networks. *Knowl. Based Syst.* **30**, 136–150 (2012)
7. Derrible, S., Kennedy, C.: Network analysis of world subway systems using updated graph theory. *Transp. Res. Rec.* **2112**(1), 17–25 (2009)
8. Fortunato, S., Hric, D.: Community detection in networks: A user guide. *Phys. Rep.* **659**, 1–44 (2016)
9. Friedkin, N.E.: Horizons of observability and limits of informal control in organizations. *Soc. Forces* **62**(1), 54–77 (1983)
10. Gao, Y., Liu, Q., Chen, G., Zhou, L., Zheng, B.: Finding causality and responsibility for probabilistic reverse skyline query non-answers. *IEEE Trans. Knowl. Data Eng.* **28**(11), 2974–2987 (2016)
11. Halpern, J.Y., Pearl, J.: Causes and explanations: a structural-model approach. Part I: Causes. *Br. J. Phil. Sci.* **56**(4), 843–887 (2005)
12. Halpern, J.Y., Pearl, J.: Causes and explanations: a structural-model approach. Part ii: Explanations. *Br. J. Phil. Sci.* **56**(4), 889–911 (2005)
13. Held, P., Krause, B., Kruse, R.: Dynamic clustering in social networks using Louvain and infomap method. In: ENIC, pp. 61–68 (2016)
14. Javed, M.A., Younis, M.S., Latif, S., Qadir, J., Baig, A.: Community detection in networks: a multidisciplinary review. *J. Netw. Comput. Appl.* **108**, 87–111 (2018)
15. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters. *Internet Math.* **6**(1), 29–123 (2009)
16. Lian, X., Chen, L.: Causality and responsibility: probabilistic queries revisited in uncertain databases. In: CIKM 2013, New York, NY, USA, pp. 349–358 (2013)
17. Malliaros, F.D., Giatsidis, C., Papadopoulos, A.N., Vazirgiannis, M.: The core decomposition of networks: theory, algorithms and applications. *VLDB J.* **29**(1), 61–92 (2020)
18. Meliou, A., Gatterbauer, W., Moore, K.F., Suciu, D.: The complexity of causality and responsibility for query answers and non-answers. *Proc. VLDB Endow.* **4**(1), 34–45 (2010)
19. Meliou, A., Roy, S., Suciu, D.: Causality and explanations in databases. *Proc. VLDB Endow.* **7**, 1715–1716 (2014)
20. Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P., Bhattacharjee, B.: Measurement and analysis of online social networks. In: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, pp. 29–42 (2007)
21. Newman, M.E.: Modularity and community structure in networks. *Proc. Natl. Acad. Sci.* **103**(23), 8577–8582 (2006)
22. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**(7043), 814–818 (2005)

23. Rossetti, G., Cazabet, R.: Community discovery in dynamic networks: a survey. *ACM Comput. Surv.* **51**(2) (2018)
24. Sharan, R., Ulitsky, I., Shamir, R.: Network-based prediction of protein function. *Mol. Syst. Biol.* **3**, 88 (2007)
25. Shen, H.W.: *Community Structure of Complex etworks*. Springer Science & Business Media (2013)
26. Tripathi, B., Parthasarathy, S., Sinha, H., Raman, K., Ravindran, B.: Adapting community detection algorithms for disease module identification in heterogeneous biological networks. *Front. Genet.* **10**, 164 (2019)
27. Wang, Z., Wang, C., Ye, X., Pei, J., Li, B.: Propagation history ranking in social networks: a causality-based approach. *Tsinghua Sci. Tech.* **25**(2), 161–179 (2020)
28. Wilber, A.W., Doye, J.P., Louis, A.A., Lewis, A.C.: Monodisperse self-assembly in a model with protein-like interactions. *J. Chem. Phys.* **131**(17), 11B602 (2009)
29. Zachary, W.W.: An information flow model for conflict and fission in small groups. *J. Anthropol. Res.* **33**(4), 452–473 (1977)
30. Zarayeneh, N., Kalyanaraman, A.: A fast and efficient incremental approach toward dynamic community detection. In: *Proceedings of the ACM/IEEE International Conference on Advances in Social Networks Analysis and Mining*, pp. 9–16 (2019)



A Pledged Community? Using Community Detection to Analyze Autocratic Cooperation in UN Co-sponsorship Networks

Cosima Meyer¹(✉) and Dennis Hammerschmidt²

¹ Mannheim Centre for European Social Research, University of Mannheim,
68159 Mannheim, Germany

cosima.meyer@uni-mannheim.de

² University of Mannheim, 68159 Mannheim, Germany

dhammers@mail.uni-mannheim.de

<http://cosimameyer.rbind.io>

<http://dennis-hammerschmidt.rbind.io>

Abstract. Autocratic cooperation is difficult to study. Democratic states usually disfavor autocratic cooperation partners because they are perceived as less reliable and do not sign agreements with them. While it is challenging to capture autocratic cooperation with traditional approaches such as signed alliance treaties, co-sponsorship at the United Nations General Assembly (UNGA) offers a valuable alternative. UNGA co-sponsorship is less binding than alliances, allowing states to cooperate more freely with one another. What is more, states are required to choose cooperation partners actively. This allows us to study how autocracies cooperate in the international system at a venue that overcomes common restrictions to autocratic cooperation. We construct co-sponsorship networks at the UNGA and use the Leiden algorithm to identify community clusters. Our multiclass random forest classification model supports our assumption and shows that regime type is associated with cooperation clusters in UNGA co-sponsorship networks.

Keywords: Social network analysis · Community detection · Machine learning · Autocratic cooperation

1 Motivation

Who cooperates with whom and why in international relations? Most of what we know about states' cooperative behavior is based on studies that focus on cooper-

Due to the random nature of surnames, the authors change the author's order on a paper basis. The authors would like to thank Brett Ashley Leeds and Nikolay Marinov for invaluable input on earlier versions of this research as well as the participants at the CoMeS 2020 conference and three anonymous reviewers. This research has been supported by the University of Mannheim's Graduate School of Economic and Social Sciences funded by the German Research Foundation.

ation among democracies [29]. These studies primarily emphasize democracies' unique domestic institutional characteristics that make them favorable cooperation partners [23]. Accordingly, democracies' high accountability for policy actions, their low flexibility to change policy decisions, and the high transparency in their policy-making process increase their costs for breaking an agreement and thereby ensure potential cooperation partners that the cooperation agreement will likely be upheld. In other words, cooperation agreements with democracies are less likely to fail, given that democracies suffer political costs from breaking an agreement [23]. This means that states—both democracies and autocracies—should prefer to cooperate with democracies. However, given their institutional similarity, democracies usually prefer to cooperate only with other democracies as they can expect similar costs for broken agreements from their cooperation partner. A result is a large number of studies that find support for democratic cooperation [23].

Autocratic cooperation, by contrast, is much less studied in the literature. Autocracies usually face more difficulties convincing potential partners that they will stick to an agreement given that their domestic institutional characteristics have relatively small costs associated with breaking an agreement [23]. As some studies show, autocracies can even benefit from breaking an agreement, especially when their costs for breaking it are relatively low [23]. While this seems to imply that autocracies would not cooperate much in the international system, several examples suggest otherwise [34]. We see that, over time, autocratic states such as Cuba, Iran, Iraq, North Korea, Sudan, Syria, and Venezuela flock together in an attempt to oppose Western liberalism [3, p. 48] and that they also cooperate among each other.

Given that almost half of all states in the world are autocracies¹, it is surprising that only a few studies so far have focused on the cooperative behavior of autocracies in the international system. Our paper addresses a central shortcoming in the study of autocratic cooperation: In traditional venues (e.g., military alliances), autocracies are less likely to find cooperation partners given that the outcome of cooperation (e.g., alliance treaties) are formalized and publicly announced agreements that require high reliability. We propose the United Nations General Assembly (UNGA) as an alternative venue to study autocratic cooperation. Given that co-sponsorship at the UNGA is non-binding and that states are required to actively choose their cooperation partners, we expect autocracies to increasingly cooperate with other autocracies when co-sponsoring resolutions at the UNGA.

¹ This estimate is based on the Polity IV score [28] to measure the share between democratic and autocratic between from 1989 and 2017.

2 Autocratic Cooperation – What We Know and What We Do Not Know

2.1 Scientific Background and Theoretical Argument

International cooperation is a concept of high scientific interest in the field of International Relations. Most basically, cooperation requires states to “adjust their behavior to the actual or anticipated preferences of others, through a process of policy coordination” [21, p. 51]. In other words, states need to negotiate with each other to reach an agreement. This implies one important aspect of cooperation: The existence of a common ground for negotiation.

A common ground for negotiations requires that all negotiation partners understand potential risks and securities that come with future cooperation. That is, before states cooperate, they weigh potential costs and benefits that are associated with specific cooperation partners [23]. Similarities in states’ domestic institutional structure serve as an important guideline here [23]. Despite the strong focus on favorable democratic institutional characteristics for international cooperation [29], it is more generally the similarity of domestic institutional characteristics that provides a common ground for negotiation [23]. While research focuses primarily on democratic cooperation, autocratic cooperation with similar domestic structures appears to be equally likely [23]. We thus argue that states with similar institutional characteristics are more likely to cooperate with each other.

Some factors may further facilitate a common ground for negotiations. We know from research that past behavior and accountability are particularly important for future cooperation. A good determinant of cooperation partners’ accountability and reliability can be derived from past cooperation patterns in the form of alliances [9, 24]. Regional similarity can further boost cooperation. At the UN, the regional groups are of particular importance for both states’ voting behavior and co-sponsorship behavior [22, 37].

2.2 The Missing Piece of the Puzzle

These arguments are not new. The importance of institutional characteristics and past cooperation behavior has been studied for decades. Yet, in the context of autocracies, they are rather difficult to analyze. In particular, this is because almost all studies on international cooperation use highly formalized treaties and official agreements (such as military alliances or trade agreements) to identify states’ cooperative behavior [19]. Due to autocracies’ unfavorable domestic institutional characteristics, as described above, states, in general, are less likely to sign a treaty with an autocracy, fearing that their autocratic partner will not uphold the agreement [23].² The result is that there is a systematic bias

² As [23] shows that this holds both for democracies and autocracies with similar concerns over failed agreements with other autocracies.

This democratic bias is also visible in terms of studies that analyze international cooperation where the almost exclusive focus is on democracies and autocracies are, at best, seen as a residual category [29].

in favor of democratic cooperation, and traditional approaches that focus on formalized treaties cannot adequately capture autocratic cooperation. To overcome this problem, we turn to co-sponsorship data on United Nations General Assembly (UNGA) resolutions.

3 Our Approach: Co-sponsorship Networks of UNGA Resolutions

UNGA resolutions are particularly useful to capture autocratic cooperation. On the one hand, UNGA resolutions are—despite their non-binding nature—highly valued in the international system. They carry a high symbolic weight that offers states international support and legitimacy for their actions [6]. The importance of legitimacy gained through UNGA resolutions can best be seen by numerous examples where states offer financial incentives to buy votes of other states to support their policy position [10]. On the other hand, UNGA resolutions are less formalized than alliances or trade agreements, and the costs for failed agreements are rather low. In other words, the low costs from broken UNGA agreements are less likely to impact states’ decisions to cooperate with each other. We argue that all UN member states can theoretically be seen as potential cooperation partners, and restrictions based on similar threats or intentions to signal strong cooperation, as it is the case for alliances, are less prevalent. This allows us to analyze autocratic cooperation without a democracy bias in our data.

Co-sponsorship of UNGA resolutions is particularly useful to study international cooperation for two reasons. First, the initial draft of a co-sponsored resolution lists only those states that were actively approached for cooperation by other states. The lead sponsor uses the internal e-deleGATE portal at the UN to invite other states to co-sponsor [31]. This allows us to analyze both democracies’ and autocracies’ choice processes efficiently and identify with whom they prefer to cooperate. If the regime type of a state is an important criterion for selecting cooperation partners, we should identify this using UNGA co-sponsorship data.³

Thus, our goal is to estimate autocratic cooperation using UNGA co-sponsorship resolutions and identify the extent to which autocracies cooperate with each other. In the following, we will outline our methodological approach in greater detail and discuss our findings in the context of both our theoretical expectations and the greater literature on autocratic cooperation. We conclude with prospects for further research.

³ Lead sponsor(s) decide on potential co-sponsors by balancing the required weight on the draft necessary to get the resolution passed and the policy positions of co-sponsors [20]. Hence, we do not expect only to find pure autocratic and pure democratic resolutions but to observe interesting patterns of cooperation across regime types as well.

4 Method

We use data from [22] to construct networks from states' UNGA co-sponsorship behavior between 1979 and 2014. To account for the relative amount that states co-sponsor in a given year⁴ we convert the $n \times n$ adjacency matrix of annual co-sponsorships for n states into an agreement matrix [1].⁵ This results in 36 weighted and directed network, one for each year in our time period. Table 1 shows an example of one agreement matrix for states' co-sponsorship behavior in 1985 where Austria (AUT) co-sponsors 100% of its resolutions with Australia (AUS), but Australia co-sponsors only 79% of its resolutions with Austria. We use these values as weights and thus receive one weighted and directed network of states' co-sponsorship behavior for each year in our sample.

Table 1. Agreement matrix of co-sponsoring at the UNGA in 1985 (first 5 rows and columns)

	AUS	AUT	BGD	BRB	CAN	...
AUS	1.00	0.79	0.61	0.35	0.79	...
AUT	1.00	1.00	0.45	0.45	0.77	...
BGD	0.71	0.41	1.00	0.71	0.71	...
BRB	0.58	0.58	1.00	1.00	0.58	...
CAN	1.00	0.77	0.77	0.45	1.00	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮

To identify community clusters of states' co-sponsorship behavior in our weighted and directed networks, we use the Leiden algorithm with Reichardt and Bornholdt's Potts partition algorithm [32, 33]. Leiden is an extension to the Louvain algorithm that guarantees connectivity and, what is most important for our case, allows us to find communities in networks that are both weighted *and* directed [36].

In a second step, we use the resulting community clusters as a target variable in a multiclass random forest model to explain the extent that the regime type explains the resulting community clusters of states' co-sponsorship behavior. It is important to note that Leiden's community clustering does not provide any meaningful attributes for each community. This means that the first community

⁴ We expect to observe differing numbers of co-sponsorships across countries. The number of co-sponsorships can, for instance, be based on the available resources of a state where larger states with much personnel at the UN headquarters are better able to engage in negotiations and co-sponsorship discussions than those that have only a limited number of staff available [31].

⁵ Not all states are UNGA members across our 36-year period and not all UNGA members co-sponsor resolutions in each year. Hence, the number of states can vary from one year to another.

in year t is not necessarily equal to the first community in year t_{+1} . As a result, we estimate 36 separate random forest models for each year to predict states' community membership based on the following variables.

We categorize states as democratic or autocratic based on data by Polity IV [28]. Polity IV is a 21-point scale measure ranging from -10 to $+10$. We are conservative with our democracy measure as we code countries being democratic if they score $+6$ or higher and autocratic otherwise. We further expect other variables to be associated with the clustering of states. First, we expect that states are more likely to co-sponsor resolutions following their alliance behavior [26] and membership in the regional groups at the UN [22].⁶ We further consider other economic and political factors such as states' trade behavior, GDP per capita, population size, the official religion, the Human Development Index, and post-conflict environments. All these variables are included in the ATOP, the Quality of Government, and the UCDP data set [25, 26, 35, 38] and used as control variables in our multiclass random forest model.⁷

One particular aspect of the UNGA is that the UN has always been used by states to communicate and advocate their domestic interests [5]. In particular, (new) states in dire need for (financial) support use(d) the UNGA to seek support. Post-conflict countries fall into this category [7, 8, 17], and we can think of two possible scenarios of how post-conflict countries might behave at the UNGA. Post-conflict states can either strategically seek for similar cooperation partners or prefer countries they perceive as big players to be their best choice.⁸

5 Results

Our results support our argument that regime type is important for international cooperation and that autocracies increasingly cooperate with each other. Figure 1 shows the distribution of democracies and autocracies for each cluster in the network in a given year. The more balanced the distribution is, i.e., the closer it is to the horizontal 50% line, the less clearly separated clusters are based on states' regime type. We find that throughout all years, we have clear

⁶ The literature shows that states usually vote in voting blocs that broadly reflect the regional groups that they belong to [4].

⁷ All variables are present across the entire observation period except of the official religion provided by the Bar-Ilan University and the Human Development Index, which are only available starting 1990 [35].

⁸ We determine the beginning of a post-conflict period by the calendar year when the UCDP/PRIOD data mark the respective country as not being in conflict [2, 14]. This event occurs once the threshold of the considered conflict is below 25 battle-related deaths [16]. If there are multiple overlapping conflict periods in a geographical country, we combine them into one single conflict period [2, 12].

Due to the limited scope of this paper, we will only discuss selected variables in the result section but include all variables in our analysis to control for confounding factors. We represent the overall mean of these features in Fig. 3.

trends for community clusters that capture either predominantly democratic or predominantly autocratic states.⁹

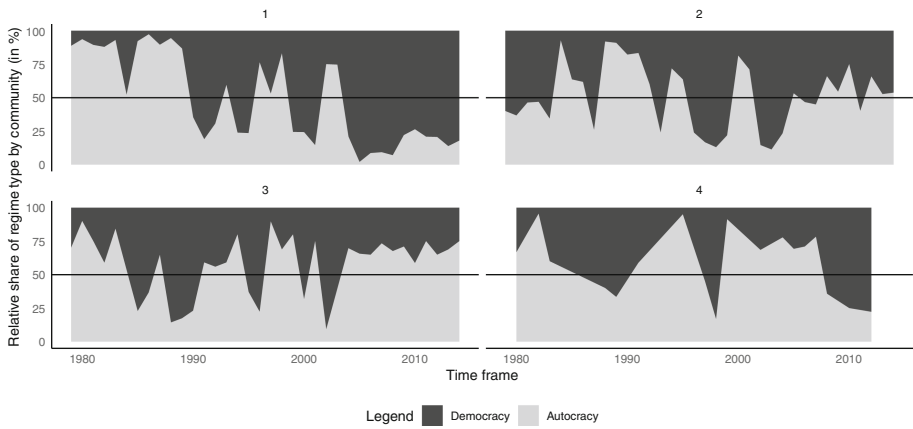


Fig. 1. Relative share of regime type (in %) by community 1–4 over time. The graph shows the distribution of democracies and autocracies in communities 1–4. The numbers associated with the communities do not contain any meaningful information on each cluster’s type, and clusters can be reshuffled for each year. The less balanced the shares (i.e., the more different from the 50% line), the more homogeneous a community is. We observe that over time communities are highly homogeneous for our entire time frame.

Figure 2 plots the network for states’ co-sponsorship at the UNGA in 1985 as an example with information on each state’s regime type and their identified community cluster using the force-directed layout algorithm by Fruchterman and Reingold [13]. Interestingly, while democracies compose one large group of states with the large majority of Western states in it (cluster 3), autocracies appear to be divided into three sub-groups. This follows the arguments in the literature. Democracy is perceived as a strong, cohesive factor that groups countries with similar institutional characteristics together. When it comes to cooperation, autocracies are not a uniform group but consist of different sub-groups that require a more disaggregated consideration [29]. We observe this in our network as well. For instance, cluster 4 consists of primarily Middle Eastern countries – all autocracies with largely similar institutional characteristics. By contrast, cluster 2 features (ex-)socialist countries such as the German Democratic Republic, Cuba, or Venezuela alongside predominantly autocratic Latin American countries.

⁹ Given that our community detection does not contain any meaningful information on the type of each cluster, as described above, and that clusters can be reshuffled for each observation year, we can observe fluctuations in the distribution of regime types over time. Put differently, our clusters only describe communities with different regimes but do not meaningfully label each cluster across the entire time period.

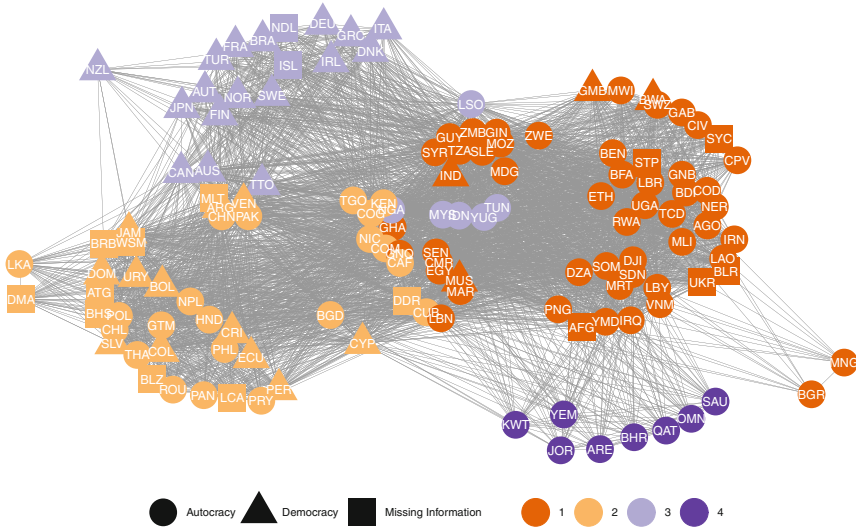


Fig. 2. Co-sponsorship network at the UNGA in 1985. The graph shows the co-sponsorship network at the UNGA in 1985. The vertex shapes represent the regime type and colors represent communities (1–4). We observe that similar regime types are clustered in communities.

Our findings for states’ cooperative behavior hold across several years and networks in our analysis and thereby reiterate previous findings of the heterogeneous nature of autocracies [29]. We believe that states’ co-sponsorship networks at the UNGA provide fruitful insights into autocratic cooperation behaviors that are difficult to study in other environments and with approaches that are not based on states’ networked behavior.

To investigate our findings of autocratic cooperation more systematically across the entire 36 years in our time frame, Fig. 3 plots the variable importance of our main determinants for co-sponsorship over time.¹⁰ These results are derived from a multiclass random forest classification. For this classification, we consider only complete cases and filter variables with zero variance, one-hot encode the dummy variables for regime and region to contain the explanatory power for our variables of interest, and normalize all non-nominal variables in our data set.¹¹

The results further support our argument and show the importance of regime type for predicting states’ membership in different co-sponsorship communities with the dotted line as a reference that indicates the mean importance across all

¹⁰ For our variable importance, we use permutation importance, which describes the difference between the prediction accuracy in the OOB observations and the prediction accuracy after randomly shuffling one single column in our data frame.

¹¹ We normalize the variables to achieve faster convergence of our models.

features.¹² Over time, the regime type of states is a consistently strong variable for the community clusters of states.¹³ In line with previous research, we further observe that both alliances and regions play an important role when understanding cluster formations [22, 26]. Both features become more important toward the end of the Cold War and experience decreasing importance during the 1990s before becoming relevant factors again at the beginning of the 2000s. However, it is interesting that while alliances experience a revival in the last period of our sample, regions are consistently decreasing in their variable importance over time. We suspect that this resembles the increasing globalization and the detachment from regional cooperation partners that can also be observed elsewhere. Post-conflict periods yield only limited importance. One reason might be that post-conflict periods tend to be relatively short (on average seven years) [15] and only occur in a small fraction of our entire sample. Moreover, post-conflict periods are rather clustered in the Americas and across the African continent. This finding might further indicate that post-conflict states instead seek to cooperate across regime types and clusters, potentially in an attempt to attract support from a wider audience. However, more research is needed to substantiate this point.

For robustness, we further estimate models with different specifications of our main independent variable regime type. Instead of a dichotomous measure of regime type, we use the original Polity IV measure and the Freedom House Index as alternative measures of the regime type and receive similar results.¹⁴ This gives us confidence that our results hold across different specifications of our model.

6 Discussion

Based on our analysis, we conclude that regime type is an important factor and that states' co-sponsorship and cooperation behavior at the UNGA correlates with the regime type. We show that both democracies and autocracies are more likely to co-sponsor resolutions at the UNGA and support previous findings that the variations in autocracies' institutional characteristics are associated with different cooperation partners. Moreover, we show that UNGA co-sponsorship networks are valuable resources to learn about international cooperation and find

¹² The mean includes alliances, states' trade behavior, GDP per capita, the population size, political regime type as well as dummies for post-conflict periods and regions.

¹³ In general, our model achieves a mean weighted AUC score of 0.81 across our models [18]. We can only calculate weighted, multiclass AUC scores for 34 out of 36 models; the remaining two models, however, achieve binary AUC scores of 0.83 and 0.97 for the years 1984 and 2003, respectively.

¹⁴ Freedom House is often referred to as an alternative measure of regime type that. Polity IV primarily focuses on the constitutional components of the regime. In contrast, Freedom House emphasizes civil liberties and political rights and makes the Freedom House Index thus more suitable for specific regions, e.g., sub-Saharan Africa [11, 30].

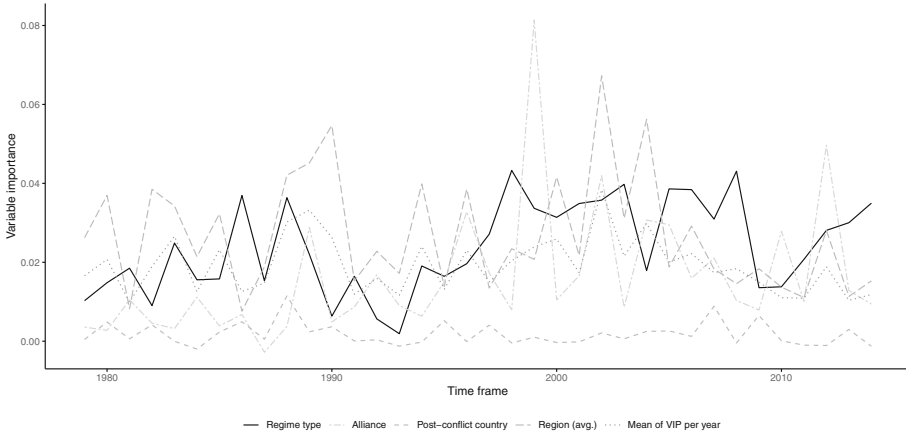


Fig. 3. Variable importance over time. The graph shows the importance of a subset of variables from the random forest model across the entire time frame. The focus is on the variable importance score of regime type, alliance, post-conflict, and an average across all five regions in the UNGA. For reference, the mean variable importance score for all features in a given year is included (dotted line). We observe that regime type is a constant and important variable for predicting the community clusters identified using Leiden across all years.

clear clusters of cooperative behavior that are in line with historical developments and insights from previous work on international relations. In particular, our findings regarding the spikes for alliances and regional patterns support previous work on international cooperation and allow for further studies on these developments. For instance, the development of cooperation patterns following the end of the Cold War and during the post-9/11 period can be observed in our results. More detailed analyses are needed here to disentangle the role of these aspects, also concerning states' regime types. One potential extension of our study might be to analyze the impact of regime transitions on states' cooperative behavior in more detail. In other words, do recently autocratized states cooperate more with fellow autocracies, or do they carry-over their cooperative behavior with democracies? These and other related questions become more important, given a third wave of autocratization that we might be currently observing [27].

Our paper shows the importance of autocratic cooperation in the field of International Relations. In particular, we highlight the possibilities that co-sponsoring networks at the UNGA offer to study further states' cooperation and behavior at the international stage.

Replication Material

The replication code can be accessed on bit.ly/replication-pledged-community.

References

1. Alemán, E., Calvo, E., Jones, M.P., Kaplan, N.: Comparing cosponsorship and roll-call ideal points. *Legislat. Studi. Q.* **34**(1), 87–116 (2009)
2. Appel, B.J., Loyle, C.E.: The economic benefits of justice: post-conflict justice and foreign direct investment. *J. Peace Res.* **49**(5), 685–699 (2012)
3. Bailey, M.A., Voeten, E.: A two-dimensional analysis of seventy years of United Nations voting. *Public Choice* **176**(1–2), 33–55 (2018)
4. Ball, M.M.: Bloc voting in the general assembly. *Int. Org.* **5**(1), 3–31 (1951)
5. Baturo, A., Dasandi, N., Mikhaylov, S.J.: Understanding state preferences with text as data: introducing the UN General Debate corpus. *Res. Polit.* **4**(2), 1–9 (2017)
6. Carter, D.B., Stone, R.W.: Democracy and multilateralism: the case of vote buying in the UN general assembly. *Int. Org.* **69**, 1–33 (2016)
7. Collier, P.: *The Bottom Billion*. Oxford University Press, Oxford (2008)
8. Collier, P., Hoeffler, A., Söderbom, M.: Post-conflict risks. *J. Peace Res.* **45**(4), 461–478 (2008)
9. Crescenzi, M.J., Kathman, J.D., Kleinberg, K.B., Wood, R.M.: Reliability, reputation, and alliance formation. *Int. Stud. Quart.* **56**, 259–274 (2012)
10. Dreher, A., Vreeland, J.R.: Buying votes and international organizations. *cege Center for European, Governance and Economic Development Research*, vol. 123, pp. 1–38 (2011)
11. Erdmann, G.: Demokratie in Afrika. *GIGA Focus Afr.* **10**, 1–8 (2007)
12. Flores, T.E., Nooruddin, I.: Democracy under the gun: understanding postconflict economic recovery. *J. Peace Res.* **53**(1), 3–29 (2009)
13. Fruchterman, T.M., Reingold, E.M.: Graph drawing by force-directed placement. *Softw. Pract. Exp.* **21**(11), 1129–1164 (1991)
14. Garriga, A.C., Phillips, B.J.: Foreign aid as a signal to investors: predicting FDI in post-conflict countries. *J. Conflict Resolut.* **58**(2), 280–306 (2014)
15. Gates, S., Nygård, H.M., Trappeniers, E.: Conflict recurrence. *Conflict Trends* **2**, 1–4 (2016)
16. Gleditsch, N.P., Wallensteen, P., Eriksson, M., Sollenberg, M., Strand, H.: Armed conflict 1946–2001: a new dataset. *J. Peace Res.* **39**(5), 615–637 (2002)
17. Hammerschmidt, D., Meyer, C.: Money makes the world go frowned. Analyzing the impact of Chinese foreign aid on states’ sentiment using natural language processing. Working Paper (2020)
18. Hand, D.J., Till, R.J.: A simple generalisation of the area under the ROC curve for multiple class classification problems. *Mach. Learn.* **45**(2), 171–186 (2001)
19. Jackson, M.O., Nei, S.: Networks of military alliances, wars, and international trade. *Proc. Natl. Acad. Sci.* **112**(50), 15277–15284 (2015)
20. Jacobsen, K.: Sponsorships in the united nations: a system analysis. *J. Peace Res.* **6**(3), 235–256 (1969)
21. Keohane, R.O.: *After Hegemony: Cooperation and Discord in the World Political Economy*. Princeton University Press, Princeton (2005)
22. Lee, E., Stek, P.E.: Shifting alliances in international organizations: a social networks analysis of co-sponsorship of UN GA resolutions, 1976–2012. *J. Contemp. Eastern Asia* **15**(2), (2016)
23. Leeds, B.A.: Domestic political institutions, credible commitments, and international cooperation. *Am. J. Polit. Sci.* **43**(4), 979–1002 (1999)

24. Leeds, B.A.: Alliance reliability in times of war: explaining state decisions to violate treaties. *Int. Org.* **57**(4), 801–827 (2003)
25. Leeds, B.A.: Alliance treaty obligations and provisions (ATOP) codebook (2018). <http://www.atopdata.org/uploads/6/9/1/3/69134503/atopcodebookv4.pdf>. Accessed 04 Apr 2020
26. Leeds, B.A., Ritter, J., Mitchell, S., Long, A.: Alliance treaty obligations and provisions, 1815–1944. *Int. Interact.* **28**(3), 237–260 (2002)
27. Lührmann, A., Lindberg, S.I.: A third wave of autocratization is here: what is new about it? *Democratization* **26**(7), 1095–1113 (2019)
28. Marshall, M.G., Gurr, T.R., Davenport, C., Jagers, K.: Polity iv, 1800–1999: comments on Munck and Verkuilen. *Comp. Polit. Stud.* **35**(1), 40–45 (2002)
29. Mattes, M., Rodriguez, M.: Autocracies and international cooperation. *Int. Stud. Quart.* **58**(3), 527–538 (2014)
30. Moss, T.J.: African development: making sense of the issues and actors. Lynne Rienner Publishers Boulder, CO (2007)
31. Panke, D.: The institutional design of the united nations general assembly: an effective equalizer? *Int. Relat.* **31**(1), 3–20 (2017)
32. Reichardt, J., Bornholdt, S.: Detecting fuzzy community structures in complex networks with a potts model. *Phys. Rev. Lett.* **93**(21), 218701 (2004)
33. Reichardt, J., Bornholdt, S.: Statistical mechanics of community detection. *Phys. Rev. E* **74**(1), 016110 (2006)
34. von Soest, C.: Democracy prevention: the international collaboration of authoritarian regimes. *Eur. J. Polit. Res.* **54**(4), 623–638 (2015)
35. Teorell, J., Dahlberg, S., Holmberg, S., Rothstein, B., Hartman, F., Svensson, R.: The Quality of Government Standard Dataset, version Jan15 (2015). <http://www.qog.pol.gu.se>. Accessed 05 June 2015
36. Traag, V.A., Waltman, L., van Eck, N.J.: From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**(1), 1–12 (2019)
37. Voeten, E.: Data and analyses of voting in the UN General Assembly (2012). <http://papers.ssrn.com/abstract=2111149>
38. Wallensteen, P., Sollenberg, M., Eriksson, M., Harbom, L., Buhaug, H., Rød, J.K.: Armed conflict dataset codebook. version 3.0 (2004). <https://www.prio.org/Global/upload/CSCW/Data/UCDP/v3/codebook.v3.0.pdf>



Distances on a Graph

Pierre Miasnikof¹(✉), Alexander Y. Shestopaloff^{2,3}, Leonidas Pitsoulis⁴,
Alexander Ponomarenko⁵, and Yuri Lawryshyn¹

¹ University of Toronto, Toronto, ON, Canada
p.miasnikof@mail.utoronto.ca

² Queen Mary University of London, London, UK

³ The Alan Turing Institute, London, UK

⁴ Aristotle University of Thessaloniki, Thessaloniki, Greece

⁵ Laboratory of Algorithms and Technologies for Networks Analysis,
Higher School of Economics, National Research University, Nizhny Novgorod,
Russian Federation

Abstract. In this article, our ultimate goal is to transform a graph's adjacency matrix into a distance matrix. Because cluster density is not observable prior to the actual clustering, our goal is to find a distance whose pairwise minimization will lead to densely connected clusters. Our thesis is centered on the widely accepted notion that strong clusters are sets of vertices with high induced subgraph density. We posit that vertices sharing more connections are closer to each other than vertices sharing fewer connections. This definition of distance differs from the usual shortest-path distance. At the cluster level, our thesis translates into low mean intra-cluster distances, which reflect high densities. We compare three distance measures from the literature. Our benchmark is the accuracy of each measure's reflection of intra-cluster density, when aggregated (averaged) at the cluster level. We conduct our tests on synthetic graphs, where clusters and intra-cluster density are known in advance. In this article, we restrict our attention to unweighted graphs with no self-loops or multiple edges. We examine the relationship between mean intra-cluster distances and intra-cluster densities. Our numerical experiments show that Jaccard and Otsuka-Ochiai offer very accurate measures of density, when averaged over vertex pairs within clusters.

1 Introduction

When clustering graphs, we seek to group nodes into clusters of nodes that are similar to each other. We posit that similarity is reflected in the number of shared connections. Our node-to-node distances are based on this shared connectivity. Although a formal definition of vertex clusters (communities) remains a topic of debate, virtually all authors agree a cluster is a subset of vertices that exhibit a high level of interconnection between themselves and a low level of connection to vertices in the rest of the graph [7, 21–23] (we quote these authors, but their definition is very common across the literature). Consequently, clusters, subsets of strongly inter-connected vertices, also form dense induced subgraphs.

Unfortunately, cluster density is not observable prior to the actual clustering. For this reason, we want a quantity that guides the aggregation of vertices into clusters, so that they form pockets of vertices separated by smaller than average distances, pockets of highly inter-connected vertices. To this end, we compare the accuracy of various node-to-node distance measures in reflecting intra-cluster density.

2 Distance, Intra-cluster Density and Graph Clustering (Network Community Detection)

As mentioned previously, clusters are defined as subsets of vertices that are considered somehow similar. This similarity is captured by the number of shared connections and translated into distance. In our model, vertices sharing a greater number of connections are separated by smaller distances than vertices with which they share fewer connections. It is important to note here that, in our definition, distance measures similarity, not geodesic (shortest path) distance. For example, two vertices with high degrees that share an edge but no other connection have a geodesic distance of one, but are dissimilar on the basis of their connectivity. At the cluster level, smaller within-cluster distances reflect subsets of more densely connected vertices.

In this article, our ultimate goal is to transform a graph's adjacency matrix into a $|V| \times |V|$ distance matrix $D = [d_{ij}]$, where the distance between each pair of vertices is given by the element $d_{ij} (\geq 0)$. This transformation allows us to cluster using distance minimization techniques from the literature. The quadratic formulation of Fan and Pardalos [5,6] and the K-medoids technique [2] are examples of such graph clustering techniques. These formulations can then be further modified into a QUBO formulation [10]. This reformulation can then be solved using newly available purpose-built hardware which allows us to circumvent the NP-hardness of the clustering problem [1,7,9,14,23].

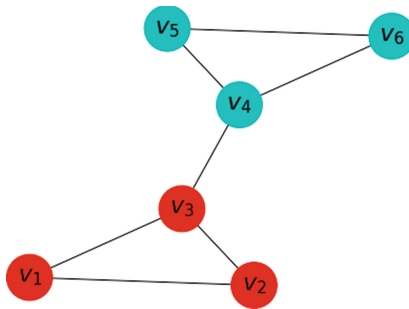


Fig. 1. Graph with two clusters (Color figure online)

To illustrate our definition of distance, we examine the graph shown in Fig. 1. The graph in that figure is arguably composed of two clusters (triangles), the red

cluster containing vertices v_1, v_2, v_3 and the cyan cluster with vertices v_4, v_5, v_6 . We observe that each cluster forms a dense induced subgraph (clique). We also note that the geodesic distance separating vertices v_1 and v_3 is equal to the geodesic distance separating v_3 and v_4 . Nevertheless, in the context of clustering, we argue that v_3 is closer, more similar, to v_1 than to v_4 .

3 Distance Measurements Under Study

We compare three different distance measurements from the literature and examine how faithfully they reflect connectivity patterns. We argue that mean node-to-node distance within a cluster should offer an accurate reflection of intra-cluster density, but move in an opposite direction. Densely connected clusters should display low mean node-node distances.

Intra-cluster density is defined as

$$K_{\text{intra}}^{(k)} = \frac{|E_{kk}|}{0.5 \times n_k \times (n_k - 1)}.$$

In this definition, $|E_{kk}|$ is the cardinality of the set of edges connecting two vertices within the same cluster ‘ k ’ and $n_k = |V_k|$ is the number of vertices in that same cluster. This ratio also represents the empirical estimate of the probability two nodes within a cluster are connected by an edge.

We then examine the relationship between mean Jaccard [12], Otsuka-Ochiai [19] and Burt’s distances [3, 7], on one hand, and intra-cluster density within each cluster, on the other. Because these distances are pairwise measures, we compare their mean value for a given cluster to the cluster’s internal density.

3.1 Embedding, Commute and Amplified Commute Distances

We begin by calling the reader’s attention to the fact this article is not about graph embedding. Here, we are not interested in a vector representation of nodes. We are only interested in the distance separating them.

We also call the reader’s attention to the fact the distance measures under consideration can all be obtained using simple arithmetic. It is precisely for this reason that we did not consider the popular “commute distance” and its corrections, like “amplified commute distance” [15, 16, 20], in this work. While these distances are known to capture cluster structure, they require matrix inversion and are very costly to compute [4]. Although some authors have found efficient approximations that circumvent the need for matrix inversion (e.g., [15]), the distances under consideration in this article are exact quantities. Exactness of the distances is a desirable feature, given our ultimate goal to use them to estimate intra-cluster density. Additionally, unlike some of the approximations in the literature, our distances have simple and intuitive interpretations.

3.2 Jaccard Distance

The Jaccard distance separating two vertices ‘ i ’ and ‘ j ’ is defined as

$$\zeta_{ij} = 1 - \frac{|c_i \cap c_j|}{|c_i \cup c_j|} \in [0, 1].$$

Here, c_i (c_j) represents the set of all vertices with which vertex ‘ i ’ (j) shares an edge.

At the cluster level, we compute the mean distance separating all pairs of vertices within the cluster, which we denote as \mathcal{J} . For an arbitrary cluster ‘ k ’ with n_k vertices, we have

$$\mathcal{J}_k = \frac{1}{0.5 \times n_k \times (n_k - 1)} \sum_{i,j=i+1} \zeta_{ij}.$$

3.3 Otsuka-Ochiai Distance

The Otsuka-Ochiai (OtOc) distance separating two vertices ‘ i ’ and ‘ j ’ is defined as

$$o_{ij} = 1 - \frac{|c_i \cap c_j|}{\sqrt{|c_i| \times |c_j|}} \in [0, 1].$$

Here too, we obtain a cluster level measure of similarity by taking the mean over each pair of nodes within a cluster. We denote this mean as \mathcal{O} . Again, for an arbitrary cluster ‘ k ’ with n_k vertices, we have

$$\mathcal{O}_k = \frac{1}{0.5 \times n_k \times (n_k - 1)} \sum_{i,j=i+1} o_{ij}.$$

3.4 Burt’s Distance

Burt’s distance between two vertices ‘ i ’ and ‘ j ’, denoted as b_{ij} , is computed using the adjacency matrix (A) as

$$b_{ij} = \sqrt{\sum_{k \neq i,j} (A_{ik} - A_{jk})^2}.$$

At the cluster level, we denote the mean Burt distance as \mathcal{B} . As with the other distances, for an arbitrary cluster ‘ k ’ with n_k vertices, it is computed as

$$\mathcal{B}_k = \frac{1}{0.5 \times n_k \times (n_k - 1)} \sum_{i,j=i+1} b_{ij}.$$

4 Numerical Comparisons

To compare the distance measures and assess the accuracy of each measure as a reflection of intra-cluster density, we generate synthetic graphs with known cluster membership, using the NetworkX library’s [11] stochastic block model generator. In our experiments, we generate several graphs with varying graph and cluster sizes and inter and intra-cluster edge probabilities. To ensure ease of readability, we only include a subset of our most revealing results.

For each test graph in the experiments below, we compute our three vertex-to-vertex distances. We then compute mean distances between nodes in each cluster and intra-cluster density. To obtain a graph-wide assessment, we then take the mean of all cluster quantities over the entire graph. Because our clusters vary in size, we ensure the well-documented “resolution limit” degeneracy [8] does not affect our conclusions by taking simple unweighted means, regardless of cluster sizes.

4.1 Test Data: Synthetic Graphs with Known Clusters

We use the stochastic block model to generate two sets of six graphs, as described in Table 1. In the first set of experiments, we vary the probability of an intra-cluster edge, an edge with both ends inside the cluster. To generate noise, we vary the size (n_k) and number of clusters (K) and as a result the total number of nodes ($|V|$). For added noise, we also set inter-cluster edge probability to 0.15. Details are shown in Table 1.

Table 1. Synthetic Graphs and their Characteristics

First set of experiments					
Graph	Intra Pr	Inter Pr	K	n_k	$ V $
G1	1	0.15	39	[38, 77]	3,641
G2	0.8	0.15	47	[38, 77]	4,703
G3	0.6	0.15	47	[38, 77]	4,326
G4	0.4	0.15	55	[38, 77]	5,386
G5	0.2	0.15	56	[38, 77]	5,557
G6	0	0.15	39	[38, 77]	3,705
Second set of experiments					
G7	1	0.15	60	[38, 77]	3,400
G8	0.8	0.15	60	[38, 77]	3,400
G9	0.6	0.15	60	[38, 77]	3,400
G10	0.4	0.15	60	[38, 77]	3,400
G11	0.2	0.15	60	[38, 77]	3,400
G12	0	0.15	60	[38, 77]	3,400

In the second set of experiments, in order to isolate the effect of intra-cluster edge probability, we keep the total number of clusters, nodes in each cluster and, consequently, total number of nodes fixed across all graphs. Although our cluster sizes vary within the graph, they are kept constant in each graph. Clusters c_1, \dots, c_K in graphs $G7, \dots, G12$ all have n_1, \dots, n_K nodes. In this experiment, we only vary intra-cluster edge probability. Details are also shown in Table 1.

4.2 Empirical Results

As mentioned earlier, we have conducted several experiments with varying graph and cluster sizes and inter and intra-cluster edge probabilities. In the interest of brevity, we only present the most illustrative subset of our results.

In our first set of experiments, we begin by observing that our results confirm intra-cluster density is a very accurate estimator of intra-cluster edge probability, under all scenarios. This observation is consistent with prior work linking densities and clustering [17, 18]. We also note that both Jaccard and OtOc distances offer a good reflection of intra-cluster density and that their change under variations in intra-cluster edge probability are in reversed lock-step with intra-cluster density. Finally, we note Burt's distance offers a poor reflection of intra-cluster density. These results are shown in Table 2.

In our second set of experiments, shown in Table 3, we observe the same relationship between distances and density. However, we also observe a factor of two reduction in the noise of both Jaccard and OtOc distances, while Burt's distance remains roughly at the same level of noise in both sets of experiments. A more detailed examination of this noise phenomenon is provided in the next section.

4.3 Noise, Sensitivity and Convergence

To better understand the sensitivity of each distance to variations in intra-cluster edge probability, we examine their asymptotic convergence. Using their definitions, we study their behavior as intra-cluster edge probability approaches 0 or 1, while keeping all else equal.

For each examination below, we define the following variables:

- P_i : probability of intra-cluster edge
- P_o : probability of inter-cluster edge
- N : total number of nodes on the graph
- n_k : number of nodes in an arbitrary cluster k , with $n_k \gg 0$
- c_i, c_j : the set of connections of two arbitrary vertices i, j in the same cluster
- A : the graph's adjacency matrix

Jaccard (and OtOc)

$$\begin{aligned} \zeta_{ij} &= 1 - \frac{|c_i \cap c_j|}{|c_i \cup c_j|} \\ &\approx 1 - \frac{P_i^2 \times (n_k - 2) + P_o^2 \times (N - n_k)}{P_i \times (n_k - 2) + P_o \times (N - n_k)} \end{aligned}$$

Table 2. First set of graph experiments (G1–G6)

	P_intra					
	0	0.2	0.4	0.6	0.8	1
Jacc						
Mean	0.919	0.918	0.912	0.898	0.879	0.841
Stdev	0.000	0.000	0.002	0.005	0.012	0.020
+1 stdev	0.919	0.919	0.914	0.903	0.891	0.862
-1 stdev	0.919	0.918	0.910	0.893	0.867	0.821
OtOc						
Mean	0.850	0.849	0.838	0.815	0.784	0.727
Stdev	0.001	0.001	0.003	0.008	0.019	0.030
+1 stdev	0.851	0.849	0.842	0.823	0.803	0.757
-1 stdev	0.849	0.848	0.835	0.807	0.765	0.697
Burt						
Mean	30.325	37.732	37.355	33.499	34.708	30.059
Stdev	0.125	0.062	0.101	0.095	0.055	0.138
+1 stdev	30.450	37.794	37.455	33.594	34.763	30.197
-1 stdev	30.200	37.671	37.254	33.404	34.653	29.921
K_intra						
Mean	0.000	0.199	0.400	0.601	0.800	1.000
Stdev	0.000	0.005	0.007	0.008	0.006	0.000
+1 stdev	0.000	0.204	0.407	0.609	0.806	1.000
-1 stdev	0.000	0.195	0.393	0.593	0.794	1.000

From this definition, we observe that

$$P_i \rightarrow 0 \Rightarrow \zeta_{ij} \rightarrow 1 - \frac{P_o^2 \times (N - n_k)}{P_o \times (N - n_k)}$$

$$P_i \rightarrow 1 \Rightarrow \zeta_{ij} \rightarrow 1 - \frac{(n_k - 2) + P_o^2 \times (N - n_k)}{(n_k - 2) + P_o \times (N - n_k)}.$$

The main observation here is that while the actual Jaccard distance depends on the number of nodes in each cluster and the total number of nodes on the graph, its variation remains in step with intra-cluster edge probability and intra-cluster density. It is this dependence on the number of nodes in each cluster and the total number of nodes on the graph that is the main source of additional variance observed in Table 2 and which is mitigated by keeping cluster sizes constant across graphs in the second set of experiments shown in Table 3. A similar argument can be made in the case of OtOc.

Table 3. Second set of graph experiments (G7–G12)

	P_intra					
	0	0.2	0.4	0.6	0.8	1
Jacc						
Mean	0.919	0.918	0.913	0.903	0.888	0.868
Stdev	0.000	0.001	0.001	0.003	0.006	0.010
+1 stdev	0.919	0.919	0.914	0.906	0.894	0.878
−1 stdev	0.919	0.918	0.911	0.899	0.882	0.858
OtOc						
Mean	0.850	0.849	0.840	0.823	0.798	0.767
Stdev	0.001	0.001	0.002	0.005	0.010	0.015
+1 stdev	0.851	0.850	0.842	0.828	0.808	0.783
−1 stdev	0.849	0.848	0.837	0.817	0.788	0.752
Burt						
Mean	29.190	29.496	29.650	29.641	29.485	29.205
Stdev	0.068	0.073	0.071	0.076	0.061	0.103
+1 stdev	29.258	29.569	29.721	29.717	29.546	29.308
−1 stdev	29.122	29.422	29.579	29.566	29.423	29.102
K_intra						
Mean	0.000	0.200	0.402	0.599	0.800	1.000
Stdev	0.000	0.011	0.015	0.011	0.011	0.000
+1 stdev	0.000	0.211	0.417	0.610	0.811	1.000
−1 stdev	0.000	0.189	0.387	0.588	0.788	1.000

Burt’s Distance

$$\begin{aligned}
 b_{ij} &= \sqrt{\sum_{k \neq i,j} (A_{ik} - A_{jk})^2} \\
 &\approx \sqrt{2 \times P_i(1 - P_i) \times (n_k - 2) + 2 \times P_o(1 - P_o) \times (N - n_k)}
 \end{aligned}$$

From this definition, we observe that

$$\begin{aligned}
 P_i \rightarrow 0 &\Rightarrow b_{ij} \rightarrow \sqrt{2 \times P_o(1 - P_o) \times (N - n_k)} \\
 P_i \rightarrow 1 &\Rightarrow b_{ij} \rightarrow \sqrt{2 \times P_o(1 - P_o) \times (N - n_k)}.
 \end{aligned}$$

On the other hand, the asymptotic behavior of Burt’s distance explains why it is a poor reflection of intra-cluster density. We see that as P_i moves toward either extreme, Burt’s distance moves toward the same quantity. It should also be noted that it is unbounded and grows with the number of nodes on the graph. In fact, as the total number of nodes increases in proportion to cluster size, the intra-cluster portion is minimized, since $(n_k - 2) \ll (N - n_k)$.

5 Our Chosen Distance

Both Jaccard and OtOc distances are very accurate reflections of intra-cluster density. Clustering by minimizing either will result in dense clusters. However, the Jaccard distance displays lower variance, in our experiments. Additionally, Jaccard similarity and its complement, the Jaccard distance, are used widely in a variety of different fields, including complex networks [4].

Because of this widespread use, lower variance and availability of pre-built computational functions, we recommend the Jaccard distance as a vertex-to-vertex distance measure for graph clustering. For example, the NetworkX library offers a Jaccard coefficient function, which we use in this work [11].

6 Metric Space and the Jaccard Distance

A metric space is a set of points that share a distance function. This function must have the following three properties:

$$g(x, y) = 0 \Leftrightarrow x = y \quad (1)$$

$$g(x, y) = g(y, x) \quad (2)$$

$$g(x, z) \leq g(x, y) + g(y, z). \quad (3)$$

In the case of the Jaccard distance, the first two properties are immediately apparent. They are direct consequences of the definitions of set operations. The third property, the triangle inequality, was shown to hold by Levandowsky and Winter [4, 13].

7 Conclusion

We show that Jaccard and Otsuka-Ochiai distances, when averaged over clusters, very accurately follow the evolution of intra-cluster density. They are both shown to vary in an opposite direction to intra-cluster density. This variation has been observed to be robust to noise from inter-cluster edge probability and variations in cluster sizes. Finally, we also show that Jaccard distance displays lower variance than Otsuka-Ochiai distance.

Our future work will focus on a study of these distances on weighted graphs. We also intend to conduct empirical comparisons to commute and amplified commute distances. We are interested in studying the statistical properties of all these distances when averaged over clusters.

Acknowledgements

- The work of Alexander Ponomarenko was conducted within the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE).
- The authors wish to thank the organizers of the 10th International Conference on Network Analysis at the Laboratory of Algorithms and Technologies for Networks Analysis in Nizhny Novgorod.

References

1. Aramon, M., Rosenberg, G., Valiante, E., Miyazawa, T., Tamura, H., Katzgraber, H.: Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. *Front. Phys.* **7**, 48 (2019). <https://doi.org/10.3389/fphy.2019.00048>
2. Bauckhage, C., Piatkowski, N., Sifa, R., Hecker, D., Wrobel, S.: A QUBO formulation of the k-medoids problem. In: Jäschke, R., Weidlich, M. (eds.) *Proceedings of the Conference on Lernen, Wissen, Daten, Analysen, CEUR Workshop Proceedings, Berlin, Germany, 30 September–2 October 2019*, vol. 2454, pp. 54–63. CEUR-WS.org (2019). http://ceur-ws.org/Vol-2454/paper_39.pdf
3. Burt, R.: Positions in networks. *Soc. Forces* **55**(1), 93–122 (1976)
4. Camby, E., Caporossi, G.: The extended Jaccard distance in complex networks. *Les Cahiers du GERAD G-2017-77* (September 2017)
5. Fan, N., Pardalos, P.: Linear and quadratic programming approaches for the general graph partitioning problem. *J. Glob. Optim.* **48**(1), 57–71 (2010). <https://doi.org/10.1007/s10898-009-9520-1>
6. Fan, N., Pardalos, P.: Robust optimization of graph partitioning and critical node detection in analyzing networks. In: *Proceedings of the 4th International Conference on Combinatorial Optimization and Applications - Volume Part I, COCOA 2010*, pp. 170–183. Springer, Heidelberg (2010). <http://dl.acm.org/citation.cfm?id=1940390.1940405>
7. Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**, 75–174 (2010)
8. Fortunato, S., Barthélemy, M.: Resolution limit in community detection. *Proc. Natl. Acad. Sci.* **104**(1), 36–41 (2007). <http://www.pnas.org/content/104/1/36.abstract>
9. Fu, Y., Anderson, P.: Application of statistical mechanics to NP-complete problems in combinatorial optimisation. *J. Phys. A Math. Gen.* **19**(9), 1605–1620 (1986)
10. Glover, F., Kochenberger, G., Du, Y.: A Tutorial on Formulating and Using QUBO Models. arXiv e-prints [arXiv:1811.11538](https://arxiv.org/abs/1811.11538) (June 2018)
11. Hagberg, A., Schult, D., Swart, P.: Exploring network structure, dynamics, and function using networkX. In: Varoquaux, G., Vaught, T., Millman, J. (eds.) *Proceedings of the 7th Python in Science Conference, Pasadena, CA, USA*, pp. 11–15 (2008)
12. Jaccard, P.: Étude de la distribution florale dans une portion des Alpes et du Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles* **37**, 547–579 (1901)
13. Levandowsky, M., Winter, D.: Distance between sets. *Nature* **234** (1971)
14. Lucas, A.: Ising formulations of many NP problems. *Front. Phys.* **2**, 5 (2014)
15. von Luxburg, U., Radl, A., Hein, M.: Getting lost in space: large sample analysis of the resistance distance. In: Lafferty, J.D., Williams, C.K.I., Shawe-Taylor, J., Zemel, R.S., Culotta, A. (eds.) *Advances in Neural Information Processing Systems*, vol. 23, pp. 2622–2630. Curran Associates, Inc. (2010). <http://papers.nips.cc/paper/3891-getting-lost-in-space-large-sample-analysis-of-the-resistance-distance.pdf>
16. von Luxburg, U., Radl, A., Hein, M.: Hitting and commute times in large random neighborhood graphs. *J. Mach. Learn. Res.* **15**(52), 1751–1798 (2014). <http://jmlr.org/papers/v15/vonluxburg14a.html>
17. Miasnikof, P., Shestopaloff, A., Bonner, A., Lawryshyn, Y.: A statistical performance analysis of graph clustering algorithms, Chap. 11. *Lecture Notes in Computer Science*. Springer Nature (June 2018)

18. Miasnikof, P., Shestopaloff, A., Bonner, A., Lawryshyn, Y., Pardalos, P.: A density-based statistical analysis of graph clustering algorithm performance. *J. Complex Netw.* **8**(3), cnaa012 (2020). <https://doi.org/10.1093/comnet/cnaa012>
19. Ochiai, A.: Zoogeographical studies on the Soleoid fishes found in Japan and its neighbouring regions-i. *Nippon Suisan Gakkaishi* **22**(9), 522–525 (1957)
20. Ponomarenko, A., Pitsoulis, L.S., Shamshetdinov, M.: Overlapping community detection in networks based on link partitioning and partitioning around medoids. *CoRR* abs/1907.08731 (2019). <http://arxiv.org/abs/1907.08731>
21. Prokhorenkova, L.O., Prałat, P., Raigorodskii, A.: Modularity of complex networks models. In: Bonato, A., Graham, F., Prałat, P. (eds.) *Algorithms and Models for the Web Graph*, pp. 115–126. Springer, Cham (2016)
22. Prokhorenkova, L.O., Prałat, P., Raigorodskii, A.: Modularity in several random graph models. *Electro. Notes Discrete Math.* **61**, 947–953 (2017), <http://www.sciencedirect.com/science/article/pii/S1571065317302238>. The European Conference on Combinatorics, Graph Theory and Applications (EUROCOMB 2017)
23. Schaeffer, S.: Survey: graph clustering. *Comput. Sci. Rev.* **1**(1), 27–64 (2007). <https://doi.org/10.1016/j.cosrev.2007.05.001>



Local Community Detection Algorithm with Self-defining Source Nodes

Saharnaz Dilmaghani¹(✉), Matthias R. Brust¹, Gregoire Danoy^{1,2},
and Pascal Bouvry^{1,2}

¹ Interdisciplinary Centre for Security, Reliability, and Trust (SnT),
University of Luxembourg, Esch-sur-Alzette, Luxembourg

{saharnaz.dilmaghani,matthias.brust}@uni.lu

² Faculty of Science, Technology and Medicine (FSTM), University of Luxembourg,
Esch-sur-Alzette, Luxembourg

{gregoire.danoy,pascal.bouvry}@uni.lu

Abstract. Surprising insights in community structures of complex networks have raised tremendous interest in developing various kinds of community detection algorithms. Considering the growing size of existing networks, *local* community detection methods have gained attention in contrast to *global* methods that impose a top-down view of global network information. Current local community detection algorithms are mainly aimed to discover local communities around a given node. Besides, their performance is influenced by the quality of the source node. In this paper, we propose a community detection algorithm that outputs all the communities of a network benefiting from a set of *local* principles and a *self-defining* source node selection. Each node in our algorithm progressively adjusts its community label based on an even more restrictive level of locality, considering its neighbours local information solely. Our algorithm offers a computational complexity of linear order with respect to the network size. Experiments on both artificial and real networks show that our algorithm gains more over networks with weak community structures compared to networks with strong community structures. Additionally, we provide experiments to demonstrate the ability of the self-defining source node of our algorithm by implementing various source node selection methods from the literature.

Keywords: Local community detection · Self-defining source node · Community structure and discovery

1 Introduction

Complex networks exhibit modular structures, namely communities, which are directly related to important functional and topological properties in various fields. They can, for example, represent modules of proteins with similar functionality in a protein interaction network [17], or affect dynamic processes of a network such as opinion and epidemic spreading [18]. Despite the various insights

and applications communities represent, they are all referred to as a densely connected set of nodes with relatively sparse links to the rest of the network. This simple definition, however, has raised great interest in discovering communities in complex networks. Numerous solutions have been proposed ever since. While most of the conventional algorithms are rooted in a top-down view obtaining the *global* information of the entire network [11, 20], others reduce the problem to a local level, by availability of a part of the network [1, 6] to find local communities of a given node(s). The existing local community detection algorithms in the literature are mostly designed to first identify a set of source nodes to initialize the community detection [2, 5, 8, 15] and then use a local community modularity to expand the communities [6, 13, 16]. The main challenges raised by these methods fall into the followings: i) the optimal result highly depends on the source node selection [5], ii) the main goal is to discover the local communities of a given set of nodes rather than all communities of a network, iii) the approaches are mostly operating in a relaxed level of locality, i.e. *local-context* the fourth level of locality [19], exploiting the information of a part of the network in the community detection process, iv) even though they appreciate a level of locality while employing the algorithm, they cannot cope with any changes in the network which is mostly the case in real-world complex networks.

Taking the above-mentioned considerations into account, we propose a community detection approach that has two main properties: First, it is operating solely based on a node and its local neighbours at a time, thus, it can belong to the *local-bounded* category, introduced by Stein et al. [19], which is one level more restrictive compared to most of the state-of-the-art approaches. Secondly, it does not depend on any auxiliary process of source node selection. Instead, it is exploiting a self-defining source node that can adapt based on the local neighbourhood knowledge. Our algorithm progressively iterates over the discovered part of the network allowing each node to decide on joining one of the neighbour communities or even create a new community. We define a community influence degree employing topological measures [10] to identify the community influence of each node. The metrics is used to guarantee a hierarchical community structure centralized by high-degree nodes. We, then, perform a local modularity measure to label each node's community. This way, our algorithm addresses the challenges raised by the previous algorithms by proposing a local approach based on a self-defining source node.

The remainder of this paper is organized as follows. In Sect. 2 we review some of the state-of-the-art in local community detection. Section 3 defines the notations and concepts that are used in the rest of the paper. In Sect. 4 we present our proposed algorithm in detail. Next, in Sect. 5 we evaluate our local community detection by using artificial and real datasets. Finally, Sect. 6 summarizes and concludes the paper.

2 Related Work

Many of *local-context* community detection algorithms are founded on this assumption that the global knowledge of the network is not available, there-

fore, the community structure measures should be in dependant of those global properties [6] such as modularity metric Q in Girvan and Newman [12] *non-local* community detection algorithm. A variety of source (i.e., seed) selection techniques are employed by *local-context* algorithms to increase the quality of communities. Some of these methods are based on the network's centrality metrics such as degree [8], others exploit similarity metrics [15] like the Jaccard score [2], while others defined new metrics, for example, node density in [5]. With all the advantages that centrality based community detection algorithms offer, they tend to give relatively poor results in dense networks and perform better in sparse networks [17]. In the next step, benefiting from a fitness function or a local community modularity, the chosen seeds are expanded. Clauset defines a local community modularity [6] as $R = \frac{\sum B_{ij}\sigma(i,j)}{\sum B_{ij}}$. It measures the ratio of the number of links within the community (i.e., internal links) to the sum of the number of all internal and external links. Luo et al. [16] have simplified the above measure and define local modularity as $M = \frac{E_{in}}{E_{out}}$, which only divides the number of internal links of a community to the number of external links. Next, Lancichinetti et al. [13] propose a fitness function as $F_c = \frac{k_{in}^c}{(k_{in}^c + k_{out}^c)^\alpha}$, where, k_{in}^c and k_{out}^c represent the internal and external links of a community c . It requires a parameter α to control the size of the communities. While the above-mentioned algorithms are considered in the *local-context* class of local algorithms [19], other algorithms perform with even more restricted local properties of a network categorized as *local-bounded*. These algorithms deploy entirely based on a node and the information from its neighbourhood. In an approach for wireless ad-hoc networks, Brust et al. [4] proposed an adaptive k -hop hierarchical community detection that performs using only neighbour local information. In another study [9] the authors proposed a community detection algorithm by giving the authority to nodes to vote for the community that they might belong to. Our local-bounded algorithm offers a change of mindset such that nodes are responsible to choose their community based on a self-identifying source selection. To expand the communities, we define a local modularity similar to Luo et al. [16] by engaging both internal and external links in the fraction.

3 Preliminaries and Notation

In this section, we introduce the preliminaries and notation that are used in the rest of this paper. We assume an undirected and unweighted network $G = (V, E)$, where V and E represent the set of nodes and the set of links, respectively. Our goal is to discover a set of all communities $C = \bigcup c_i$, such that each node $v \in V$ belongs only to one community. A *good* community is achieved if all nodes within a community are densely intra-connected, in other words, implying that the local modularity of each community is maximized. Besides, we construct a community in a hierarchical structure in such a way that nodes with a higher degree are pushed towards the center of the community whereas the lower degree nodes stay

close to the border of the community. We aim to find all communities of a network by allowing each node to adjust its community label given its local neighbours, $\Gamma(v)$, and their properties at a time. We exploit a set of measures adopted from the network structure to assure that each node belongs to a community at the end of the execution time.

Definition 1 (*Community influence degree*). Each node is influenced by its surrounding communities. To quantify this impact, we define $\lambda(v)_{c_i}$ to show the level of impact from node v with community label c_i to its neighbours, as follows:

$$\lambda(v)_{c_i} = \frac{k_v}{hl}, \quad (1)$$

where k_v is the degree of v (i.e. the number of nodes in $\Gamma(v)$), and hl shows the hierarchy level of v in its community. In a nutshell, hl represents the hop distance from the source node in the community. The value is 1 for source nodes, showing the first layer of the hierarchy (i.e., seed node) and increases by per hop-distance towards the border of the community. The intuition behind this measure is that a node is more likely to be in the same community as another node if the following node is closer to the source of the community and has a higher degree. Thus, we indicate the *strength* of a member in a community with a high $\lambda(v)$ value showing the high degree and low hierarchy level of that node.

Definition 2 (*Local community modularity*). It defines the degree of a node contributing to a candidate community c_i . It is measured by the following equation:

$$\mu(v)_{c_i} = \frac{E_{in} - E_{out}}{E_{in} + E_{out}} = 2 \frac{E_{in}}{E_{in} + E_{out}} - 1, \quad (2)$$

where E_{in} is the number of edges from node v towards the community c_i , E_{out} represents the outwards of node v . Therefore, $k_v = E_{in} + E_{out}$ is the total number of edges of v or simply the degree of node v . In other words, the local community modularity explains a membership degree for a given community. It represents the link ratio of those neighbours of v within a community minus the number of those outside the community, normalized by the degree of v . The value can vary in the range of $(-1, 1]$. It takes a negative value if it does not have any connection to the community c_i and positive if the majority of its links are toward the community.

4 Self-defining Local Community Detection

We design an iterative bottom-up approach allowing each node to take a decision of joining a community independently. Our algorithm discovers the whole network starting from a given node and its local neighbours, therefore, it performs in a restricted level of locality (i.e. local-bounded). The algorithm converges when all nodes agree with their community labels. We assume a hierarchical structure for each community by encouraging high degree nodes towards the center of the

community and nodes with a lower degree to the borders while maximizing the local modularity defined in Eq. 2. To forge a hierarchical structure, we adjust the hop-distance hl , and in the meantime, we update each node's community influence degree $\lambda(v)$ as defined in Eq. 1. The metric is considered as a level of attraction to encourage a node towards a community. On the other hand, to extend communities or to prevent emerging large communities we initially filter communities by measuring the local modularity from Eq. 2.

Algorithm description. The general structure of the proposed local approach to detect communities of a network is described in Algorithm 1. To extend the communities we define a set of principles that are explained in Algorithm 2. The procedure starts by initializing the node list R (line 1), that records visited nodes and their neighbours. As a first-time-visited node in the list, the community label cl and hierarchy level hl of the node will be initialized to its node ID and a constant value HL , respectively (line 2–3). We chose HL to be 4 initially, however, it can be any value larger than 1. The next step is to adjust the node's hl value, its value will be reduced if it has the highest degree compared to its neighbours (line 6–7). Afterwards, the community influence degree $\lambda(v)$ and the local modularity $\mu(v)$ is calculated (line 9–10). To update both hl and cl of v , we input the node through some principles defined in Algorithm 2 (line 11). Besides, the list R will be updated by the neighbours of node v . Finally, if all nodes come to an agreement such that no further changes occur, the algorithm will converge and stop. Extracting the cl of all nodes in R results in obtaining all communities of G . A set of principles is defined in Algorithm 2 to decide the corresponding community of the node v . First, choosing the common community

Algorithm 1. Adaptive local community detection

Input: Node v , and $\Gamma(v)$

Output: C set of communities

Initialisation:

- 1: $R \leftarrow v$
- 2: $v.hl = HL$
- 3: $v.cl = v$

Procedure

- 4: **while** stopCondition **do**
 - 5: **for** v in R **do**
 - 6: **if** $\deg(v) > \deg(\Gamma(v))$ **then**
 - 7: $v.hl \leftarrow v.hl - 1$
 - 8: **end if**
 - 9: $v.\lambda = \lambda(v)$
 - 10: $v.\mu = \mu(v)$
 - 11: $v.hl, v.cl \leftarrow \text{Alg. 2}(v)$
 - 12: $R \leftarrow \text{update}(\Gamma(v))$
 - 13: **end for**
 - 14: **end while**
 - 15: **return** $C \leftarrow R.cl$
-

label (mc), the local modularity is calculated. If $v.\mu$ was positive, v takes the same label as mc . Then, v adjust its hierarchy level by taking the minimum hl of that community and increase it by one unit as its hl value. Otherwise, if $\mu(v)$ was negative or zero, then, either v itself is selected by the neighbours to be a new community, or it will temporarily follow the best candidate among its neighbourhood.

Algorithm 2. Local community expansion

```

1:  $mc =$  common community label
2:  $bc = [u \text{ in } \Gamma(v) \text{ if } u.\lambda \text{ is } \max(\Gamma(v).\lambda)]$ 
3: if ( $v.\mu > 0$ ) then
4:    $v.hl = \min(\Gamma(v)).hl + 1$ 
5:    $v.cl = mc$ 
6: else if ( $v.\mu \leq 0$ ) then
7:   if  $v$  is  $mc$  then
8:      $v.hl = 1$ 
9:      $v.cl = v$ 
10:  else
11:     $v.hl = bc.hl$ 
12:     $v.cl = bc.cl$ 
13:  end if
14: end if

```

Computational Complexity. The complexity of the proposed algorithm, on a network of size n , and an average degree k can be estimated as follows. The outer *while* loop repeats until the algorithm has converged. The inner *for*-loop, depends on the length of R which progressively includes all the nodes from V . Starting from one node with degree k , in the worst case, R increases as follows: $\{1, k, k^2, \dots, k^m\}$, while $k^m = n$, hence, it is in the order of n and can never be more than $O(m \times n)$, with m as the number of iterations in the outer *while* loop until the convergence.

5 Experimental Analysis

In this section, we examine the performance of our algorithm with different experiments. We exploit both real-world and artificial networks that are described in Table 1. Following artificial networks, we generate various networks using the LFR benchmark algorithm [14]. The mixing parameter μ , identifies the density of the networks, i.e. the strength of the communities.

We first compare the results of the proposed algorithm on the networks from Table 1 with a set of algorithms: Louvain [3] and Fast-greedy [7], and Label Propagation Algorithm (LPA). Next, to examine the ability of self-defining source nodes of our algorithm, we implement a set of source node selection methods from the literature and combine them with our algorithm. Finally, we provide tests to validate the analytically derived low complexity of our algorithm.

Table 1. Dataset of networks used for the experiments.

Real-world networks with ground-truth								
Networks	n	k_{avg}	n_{com}	Description				
Zachery's Club	34	4.59	2	Zachary's karate club				
Football	115	10.66	12	American football game				
Dolphins	62	5.13	2	Dolphin social networks				
US Politics' Books	105	8.5	3	US Politics' Books				
Synthetic networks								
Networks	n	k_{avg}	μ	t_1	t_2	c_{min}	c_{max}	n_{com}
LFR 4000	4000	25	0.1 – 0.8	2	1.1	40	100	63
LFR 8000	8000	25	0.1 – 0.8	2	1.1	60	100	103
LFR 15000	15000	25	0.1 – 0.8	2	1.1	40	200	82

Table 2. The AMI quality metric results on the communities detected by Louvain, LPA, Fast-greedy, and our proposed algorithm (Proposed Alg.) on real-world networks. The bold values show the best results among other algorithms for each network.

Networks	Louvain	LPA	Fast-greedy	Proposed Alg.
Zachery's club	0.46	0.48	0.54	0.45
Football	0.85	0.87	0.65	0.65
Dolphins	0.49	0.59	0.55	0.88
US Politics' books	0.49	0.53	0.51	0.56

5.1 Evaluating Quality of Communities

To measure the quality of the results obtained from networks in Table 1, we calculate Adjusted Mutual Information (AMI). This metric is an adjustment of the Mutual Information (MI) score to account for chance. It accounts for the fact that the MI is generally higher for two methods with a larger number of communities, regardless of whether there is actually more information shared.

We compare the results with the above-mentioned algorithms from the literature. The resulted communities of these algorithms are, then, used as a baseline to compare the performance of our algorithm with. The results for the real-world networks are reported in Table 2 and for LFR benchmark networks are shown in Fig. 1. As shown in both results, our algorithm is comparable to the other algorithms while processing entirely based on the local information and thus, benefiting from a low complexity. The algorithm gains more when the community structure of the network becomes weaker (i.e., μ is increased). The reason is that due to the locality level, our algorithm behaves greedily in a situation where the conditions to join a neighbour community are not fulfilled, by generating new communities. Hence, it ends up with different communities than the other algorithms of Louvain and LPA, and similar to Fast-greedy.

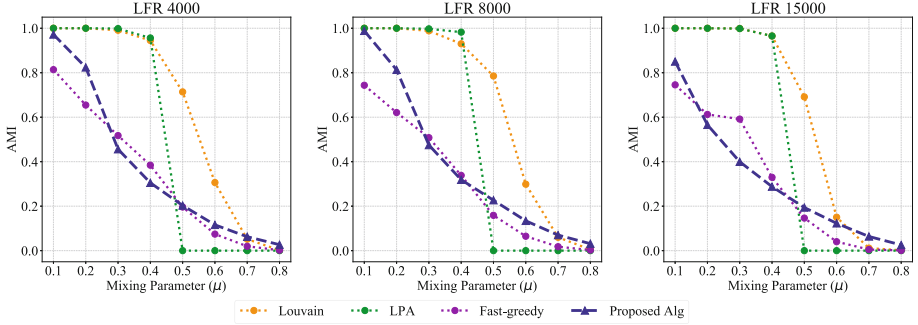


Fig. 1. AMI results on the LFR benchmark networks explained in Table 1.

5.2 Source Node Selection Analysis

Most of the existing local community detection algorithms require a source node selection before the community expansion. We implement some of the source node selection methods from the literature and develop an experiment to analyze the impact of source node selection on our algorithm. We choose different centrality and similarity scores: degree centrality [8], extended Jaccard metric [2], and node density (to find nodes with high degree, however, distant from each other) [5]. In order to be fair on choosing the best candidate nodes, we apply an outlier detection technique, Interquartile Range (IQR), to select nodes with higher scores. We then adjust the hl of these nodes to be known as the initial communities of the network and proceed as described in Algorithm 1. We evaluate the methods on an LFR benchmark network with $n = 2000$ and report the results in Fig. 2. The results show that there are no differences between the proposed algorithm (Basic) and its variations by each source node selection (e.g., Basic+Degree). As shown in Fig. 2, our method maintains a self-identifying source node selection considering node degree.

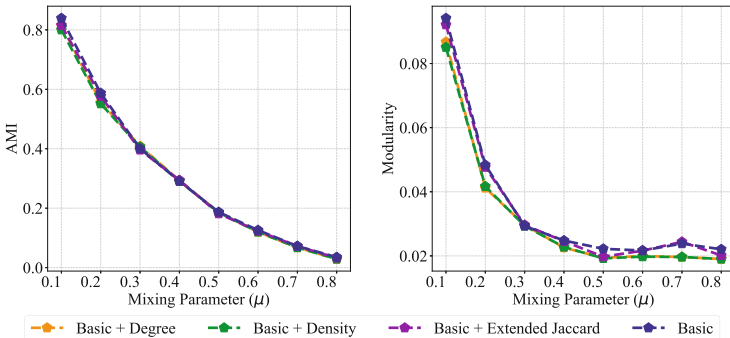


Fig. 2. Employing different source node selection methods from the literature on the bases of the proposed algorithm. The methods are examined over the LFR 2000s network exploiting AMI and Modularity measures.

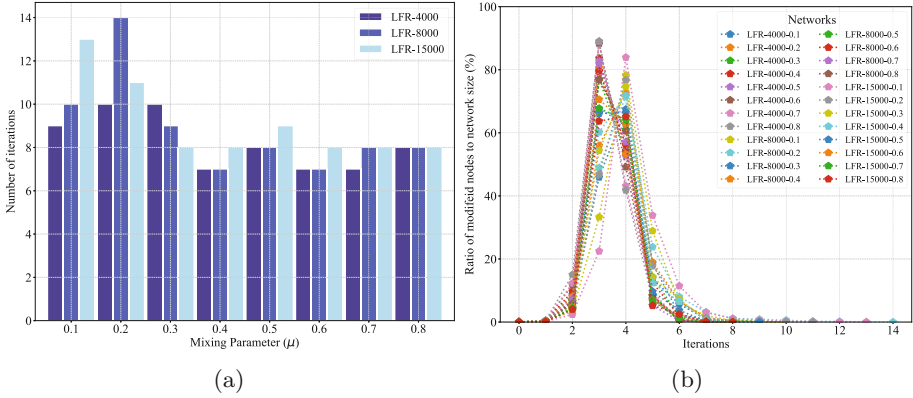


Fig. 3. The results of experiments on the convergence of the algorithm on LFR networks, (a) Bar plot of the number of iteration, (b) The percentage of the number of nodes modified per iteration.

5.3 Computational Complexity Analysis

Following the experiments on the networks provided in Table 1, we analyze both the number of iterations our algorithm requires to converge (the outer loop in Algorithm 2) and the number of nodes from the list R that were qualified to the conditions thus, are forged to change adjust their properties (i.e., hl or cl) in Algorithm 2 which are not all the nodes in R . The overall results are shown in Fig. 3. At each level of the mixing parameter (μ), from 0.1 to 0.8, for each network size, we calculated the number of iterations that the algorithm requires until convergence. As shown in Fig. 3a, the number of repetitions does not rely on the size of the network and is slightly influenced by μ that shows the organization of community structures. However, regardless of n , the proposed algorithm converges in the average number of 8.2 iterations. Furthermore, with regard to the inner loop of the algorithm, we calculate a ratio of the number of nodes that are entitled to modify in each iteration to the size of the network. According to Fig. 3b, the results reveal that the number of operations in each repetition of the algorithm has never reached n . It hits 87% of n in its maximum case which has mostly occurred from 3rd to 5th iterations. The number of modified nodes are considerably lower than the 3rd to 5th iterations that substantiates the low complexity of our algorithm.

6 Conclusion and Future Work

In this paper, we described our proposed community detection algorithm that is benefiting from a set of local principles and a self-defining source node selection to detect communities in complex networks. We developed the algorithm exploiting community influence degree and a local community modularity that are defined in this paper. The community influence degree of a node increases

if the node has a high degree and low hierarchy level in the community that is defined based on the hop-distance from the source node. This way, we shape communities in a hierarchical structure where nodes with higher degrees are towards the center of the community. Our algorithm exploits a set of local principles allowing each node to take a decision on its community label based on its neighborhoods local information. The algorithm is designed in a more restrictive level of locality compared to the current local algorithms and offers a linear order of computational complexity. We deploy extensive experiments to analyze the performance and efficiency of our algorithm. The experiments on both real and artificial networks show that the proposed algorithm performs better in networks with weak community structures compare to the algorithms that benefit from the global information of the network. Moreover, we perform experiments to validate the ability of self-defining source node selection of the our algorithm. We show that our algorithm performs independently from the source node selection methods in the literature. The experiments on the complexity of the algorithms demonstrate that, regardless of the size of the network, the algorithm converged after approximately 8 iterations, whereas, the number of nodes that are involved in the process has shown not to exceed the 87% of the whole network size. Remarkably, the locality and self-defining properties of this approach have equipped our algorithm for the future investigations on the adaptability to dynamic environments. Besides, we are planning to elaborate on the proposed approach by employing a local merging method on the output communities in order to increase the accuracy and performance of the results, while still holding the same level of the locality.

Acknowledgment. This work has been partially funded by the joint research programme University of Luxembourg/SnT-ILNAS on Digital Trust for Smart-ICT.

References

1. Bagrow, J.P., Bollt, E.M.: Local method for detecting communities. *Phys. Rev. E* **72**(4), 046108 (2005)
2. Berahmand, K., Bouyer, A., Vasighi, M.: Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes. *IEEE Trans. Comput. Soc. Syst.* **5**(4), 1021–1033 (2018)
3. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech.*
4. Brust, M.R., Frey, H., Rothkugel, S.: Adaptive multi-hop clustering in mobile networks. In: *Proceedings of the 4th International Conference on Mobile Technology, Applications* (2007)
5. Chen, Y., Zhao, P., Li, P., Zhang, K., Zhang, J.: Finding communities by their centers. *Sci. Rep.* **6**, 24017 (2016)
6. Clauset, A.: Finding local community structure in networks. *Phys. Rev.* **E72**(2), 026132 (2005)
7. Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. *Phys. Rev. E* **70**(6), 066111 (2004)

8. Comin, C.H., da Fontoura Costa, L.: Identifying the starting point of a spreading process in complex networks. *Phys. Rev. E* **84**(5), 056105 (2011)
9. Coscia, M., Rossetti, G., Giannotti, F., Pedreschi, D.: DEMON: a local-first discovery method for overlapping communities. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2012)
10. Dilmaghani, S., Brust, M.R., Piyatumrong, A., Danoy, G., Bouvry, P.: Link definition ameliorating community detection in collaboration networks. *Front. Big Data* **2**, 22 (2019)
11. Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**(3–5), 75–174 (2010)
12. Girvan, M., Newman, M.E.: Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* **99**(12), 7821–7826 (2002)
13. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. *J. phys.* **11**(3), 033015 (2009)
14. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78**(4), 046110 (2008)
15. Li, S., Huang, J., Zhang, Z., Liu, J., Huang, T., Chen, H.: Similarity-based future common neighbors model for link prediction in complex networks. *Sci. Rep.* **8**(1), 1–11 (2018)
16. Luo, F., Wang, J.Z., Promislow, E.: Exploring local community structures in large networks. In: 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006) (2006)
17. Porter, M.A., Onnela, J.P., Mucha, P.J.: Communities in networks. *Not. AMS* **56**(9), 1082–1097 (2009)
18. Stegehuis, C., Van Der Hofstad, R., Van Leeuwen, J.S.: Epidemic spreading on complex networks with community structures. *Sci. Rep.* **6**(1), 1–7 (2016)
19. Stein, M., Fischer, M., Schweizer, I., Mühlhäuser, M.: A classification of locality in network research. *ACM Comput. Surv. (CSUR)* **50**(4), 1–37 (2017)
20. Yang, Z., Algesheimer, R., Tessone, C.J.: A comparative analysis of community detection algorithms on artificial networks. *Sci. Rep.* **6**, 30750 (2016)



Investigating Centrality Measures in Social Networks with Community Structure

Stephany Rajeh^(✉), Marinette Savonnet, Eric Leclercq, and Hocine Cherifi

Laboratoire d'Informatique de Bourgogne, University of Burgundy, Dijon, France
stephany.rajeh@u-bourgogne.fr

Abstract. Centrality measures are crucial in quantifying the influence of the members of a social network. Although there has been a great deal of work dealing with this issue, the vast majority of classical centrality measures are agnostic of the community structure characterizing many social networks. Recent works have developed community-aware centrality measures that exploit features of the community structure information encountered in most real-world complex networks. In this paper, we investigate the interactions between 5 popular classical centrality measures and 5 community-aware centrality measures using 8 real-world online networks. Correlation as well as similarity measures between both types of centrality measures are computed. Results show that community-aware centrality measures can be divided into two groups. The first group, which includes Bridging centrality, Community Hub-Bridge, and Participation Coefficient, provides distinctive node information as compared to classical centrality. This behavior is consistent across the networks. The second group which includes Community-based Mediator and Number of Neighboring Communities is characterized by more mixed results that vary across networks.

Keywords: Centrality · Community structure · Influential nodes

1 Introduction

With the rapid increase of online social networks (OSNs) such as Facebook and Twitter, large amount of data is being generated daily. A valuable mining area of network data is composed when OSNs are modeled into nodes and edges. Identifying key nodes in such networks is the basis of major applications such as viral marketing [1], controlling epidemic spreading [2], and determining sources of misinformation [3]. Designing centrality measures is a main approach to quantify node influence. Numerous centrality measures exploiting various properties of the network topology have been developed [4]. Information exploited can be either in the neighborhood of the node or concerning all the topological structure of the network. The former called local centrality measures are less computationally expensive as compared to the later called global centrality measures. However,

local centrality measures, usually, aren't as much as accurate as global centrality measures. Recent works tend to combine both local and global measures [5,6]. Real-world OSNs often exhibit a community structure in which groups of nodes are closely connected to each other and sparsely connected to nodes in other communities [7,8]. Community structure has major implications on the dynamics of the network [9]. To this end, researchers have taken classical centrality measures a step further to incorporate community structure information [10–17]. Community-aware centrality measures can be divided into two groups. The former explicitly rely on the community structure. They incorporate information about the type of links in a community (intra-community links and inter-community links). The latter targets “bridges” that lie between communities without extracting the community structure information.

As classical centrality measures neglect the community structure, this raises a key question. Do community-aware centrality measures provide distinctive information about the members within OSNs when compared to classical centrality measures? Previous works have studied the relationship between classical centrality measures [18–22] and between classical and hierarchy measures [23]. Nonetheless, to our knowledge, there is no previous work on the relationship between classical and community-aware centrality measures on OSNs. To fill this gap, here, 5 classical and 5 community-aware centrality measures are used in a comparative evaluation involving 8 real-world OSN. The community structure of the networks is extracted using the Infomap [24] community detection algorithm. Then, Kendall's Tau correlation and RBO similarity are calculated on all the possible combinations between the classical and community-aware centrality measures. Two groups of community-aware centrality measures can be seen. The first group provides distinctive information when compared against classical centrality measures and is consistent across the networks under study. It includes Bridging centrality, Community Hub-Bridge, and Participation Coefficient. The second group shows varying correlation and similarity on networks. It includes Community-based Mediator and Number of Neighboring Communities.

The paper is organized as follows. Classical and community-aware centrality measures alongside basic definitions are provided in Sect. 2. The datasets and tools are provided in Sect. 3. Experimental results are discussed in Sect. 4. Finally, the conclusion and future works are provided in Sect. 5.

2 Preliminaries and Definitions

In this section preliminaries and definitions used throughout the rest of the paper are given.

- Consider a undirected and unweighted OSN as $G(V, E)$ where V is the set of nodes and $E \subseteq V \times V$ is the set of edges and $N = |V|$ is the total size of the network. Nodes represent individuals and edges represent social links between these individuals. The semantics of the social links depend on the platform of the OSN.

- Consider $A = (a_{i,j})$ as the adjacency matrix showing connectivity of the network G such that $a_{i,j} = 1$, if node i is connected to node j and $a_{i,j} = 0$, otherwise.
- Let the neighborhood of any node i be defined as the set $\mathcal{N}_p(i) = \{j \in V : (i, j) \in E\}$ at length p , where $p = 1, 2, \dots, D$. D is the diameter of G . Accordingly, two nodes are neighbors of order A^p if there's a minimal path connecting them at p steps.
- Let C be the set of communities $C = \{c_1, c_2, \dots, c_k\}$. The intra-community links are obtained from the graph G_l where all inter-community links of the nodes are removed. The inter-community links are obtained from the graph G_g where all intra-community links of the nodes are removed.

2.1 Classical Centrality Measures

Following are the definitions of the 5 most popular centrality measures used in the study.

Degree Centrality is simply the total number of connections a node has in the network. It is defined as follows:

$$\alpha_d(i) = \sum_{j=1}^N a_{ij} \quad (1)$$

where a_{ij} is obtained from A^1 , 1-step neighborhood ($p = 1$).

Betweenness Centrality captures the number of times a node falls between the shortest paths linking other node pairs. It is defined as follows:

$$\alpha_b(i) = \sum_{s,t \neq i} \frac{\sigma_i(s,t)}{\sigma(s,t)} \quad (2)$$

where $\sigma(s,t)$ is the number of shortest paths between nodes s and t and $\sigma_i(s,t)$ is the number of shortest paths between nodes s and t that pass through node i .

Closeness Centrality is the inverse of the sum of geodesic distances to every other node from a given node. It is defined as follows:

$$\alpha_c(i) = \frac{N-1}{\sum_{j=1}^{N-1} d(i,j)} \quad (3)$$

where $d(i,j)$ is the shortest-path distance between node i and j .

Katz Centrality is based on how many nodes a node is connected to and also to the connectivity of its neighbors. It is defined as follows:

$$\alpha_k(i) = \sum_{p=1} \sum_{j=1} s^p a_{ij}^p \quad (4)$$

where a_{ij}^p is the connectivity of node i with respect to all the other nodes at A^p and s^p is the attenuation factor where $s \in [0, 1]$.

PageRank Centrality quantifies a node's importance similarly to Katz centrality with an additional layer based on a random surfer. It is defined as follows:

$$\alpha_p(i) = \frac{1-d}{N} + d \sum_{j \in \mathcal{N}_1(i)} \frac{\alpha_p(j)}{k_j} \quad (5)$$

where $\alpha_p(i)$ and $\alpha_p(j)$ are the PageRank centralities of node i and node j , respectively, $\mathcal{N}_1(i)$ is the set of direct neighbors of node i , k_j is the number of links from node j to node i , and d is the damping parameter where $d \in [0, 1]$, set to 0.85 in the experiments.

2.2 Community-Aware Centrality Measures

Following are the definitions of the 5 community-aware measures of centrality used:

Number of Neighboring Communities (NNC) [11] is based on the number of communities a node can reach in one hop. For a node in community $c_k \subset C$, it is defined as follows:

$$\beta_{NNC}(i) = \sum_{c_l \subset C \setminus c_k} \bigvee_{j \in c_l} a_{ij} \quad (6)$$

where $\bigvee_{j \in c_l} a_{ij} = 1$ when node i is connected to at least one node j in community c_l .

Community Hub-Bridge (CHB) [11] assumes a node simultaneously can act as a hub and a bridge. It combines the intra-community and inter-community links by weighting the former with the community size and the latter with the number of neighboring communities. For a node in community $c_k \subset C$, it is defined as follows:

$$\beta_{CHB}(i) = h_i(c_k) + b_i(c_k) \quad (7)$$

where hub influence is given by $h_i(c_k) = |c_k| \times k_i^{intra}$ and bridge influence is given by $b_i(c_k) = \beta_{NNC}(i) \times k_i^{inter}$.

Participation Coefficient (PC) [12] is based on the intra-community and inter-community links distribution. The more the links of a node are distributed

across different communities, the higher its participation coefficient. It is defined as follows:

$$\beta_{PC}(i) = 1 - \sum_{c=1}^{N_c} \left(\frac{k_{i,c}}{k_i^{tot}} \right)^2 \quad (8)$$

where N_c is the total number of communities, $k_{i,c}$ is the number of links node i has in a given community c (can be inter-community or intra-community links), and k_i^{tot} is the total degree of node i .

Community-based Mediator (CBM) [13] takes into consideration the intra-community and inter-community ratio of a node, then it incorporates a random walker and entropy based on the ratio of the different link types. It is defined as follows:

$$\beta_{CBM}(i) = H_i \times \frac{k_i^{tot}}{\sum_{i=1}^N k_i} \quad (9)$$

where $H_i = [-\sum \rho_i^{intra} \log(\rho_i^{intra})] + [-\sum \rho_i^{inter} \log(\rho_i^{inter})]$ is the entropy of node i based on its ρ_i^{intra} and ρ_i^{inter} which represent the density of the communities a node links to (either its community or external communities), k_i^{tot} is the total degree of node i , and $\sum_{i=1}^N k_i$ is the total degrees in the network.

Bridging Centrality (BC) [10] extracts node bridges by using betweenness centrality and bridging coefficient. The bridging coefficient quantifies the proximity of a node to high degree nodes. It is defined as follows:

$$\beta_{BC}(i) = \alpha_b(i) \times \mathbb{B}(i) \quad (10)$$

where $\alpha_b(i)$ is the classical betweenness centrality of node i and $\mathbb{B}(i) = \frac{k_i^{-1}}{\sum_{j \in \mathcal{N}_1(i)} k_j^{-1}}$ is the bridging coefficient where $\mathcal{N}_1(i)$ is the set of direct neighbors of node i .

3 Datasets and Materials

In this section, the 8 real-world online social networks are briefly discussed, alongside the tools applied. Table 1 reports the basic topological characteristics of the networks. Note that the mixing parameter μ is defined as the proportion of inter-community links to the total links in a given network. It is calculated after the community structure is uncovered by the community detection algorithm.

3.1 Data

FB Ego this network (ego-facebook) is collected from participants using Facebook. Nodes represent users on Facebook and edges represent online friendships [25].

Table 1. Basic topological properties of the real-world networks. N is the total number of nodes. E is the number of edges. $\langle k \rangle$ is the average degree. $\langle d \rangle$ is the average shortest path. ν is the density. ζ is the transitivity (also called global clustering coefficient). $k_{nn}(k)$ is the assortativity (also called degree correlation coefficient). Q is the modularity. μ is the mixing parameter. * indicates the topological properties of the largest connected component of the network in case it is disconnected.

Network	N	E	$\langle k \rangle$	$\langle d \rangle$	ν	ζ	$k_{nn}(k)$	Q	μ
Retweets Copenhagen	761	1, 029	2.70	5.35	0.003	0.060	-0.099	0.695	0.287
FB Caltech*	762	16, 651	43.70	2.23	0.057	0.291	-0.066	0.389	0.410
Hamsterster*	1, 788	12, 476	13.49	3.45	0.007	0.090	-0.088	0.391	0.298
FB Ego	4, 039	88, 234	43.69	3.69	0.010	0.519	0.063	0.814	0.077
FB Politician Pages	5, 908	41, 729	14.12	4.66	0.002	0.301	0.018	0.836	0.111
FB Princeton*	6, 575	293, 307	89.21	2.67	0.013	0.163	0.090	0.417	0.365
PGP	10, 680	24, 316	4.55	7.48	0.0004	0.378	0.238	0.813	0.172
DeezerEU	28, 281	92, 752	6.55	6.44	0.002	0.095	0.104	0.565	0.429

FB Princeton this network (socfb-Princeton12) is collected from Facebook among students at Princeton University. Nodes represent users on Facebook and edges represent online friendships [25].

FB Caltech this network (socfb-Caltech36) is collected from the Facebook application among students at Caltech University. Nodes represent users on Facebook and edges represent online friendships [25].

FB Politician Pages this network (fb-pages-politician) is collected from Facebook pages. Nodes represent politician pages from different countries created on Facebook and edges represent mutual likes among them [25].

Retweets Copenhagen this network (rt-twitter-copen) is collected from Twitter. Nodes are users on Twitter tweeting in parallel to the United Nations conference in Copenhagen about climate change and edges represent retweets among the users [25].

DeezerEU this network (deezer_europe) is obtained from Deezer, a platform for music streaming. Nodes are Deezer European users and edges represent online friendships [26].

Hamsterster this network (petster-friendships-hamster) is obtained from an online social pet network hamsterster.com. Nodes represent users and edges represent friendships among them [27].

PGP this network (arenas-pgp) is obtained from the web of trust. Nodes are users using the Pretty Good Privacy (PGP) algorithm and edges represent secure information sharing among them [27].

3.2 Tools

Kendall’s Tau Correlation is used to assess the relationship for all possible combinations between classical and community-aware centrality measures. Assume that $R(\alpha)$ and $R(\beta)$ are the ranking lists of a classical centrality and a community-aware centrality, respectively. The correlation value resulted $[-1, +1]$ reveals the degree of ordinal association between the two given sets of ranks. If $R(\alpha_i) > R(\alpha_j)$ and $R(\beta_i) > R(\beta_j)$ or $R(\alpha_i) < R(\alpha_j)$ and $R(\beta_i) < R(\beta_j)$, node pair (i, j) is concordant. If $R(\alpha_i) > R(\alpha_j)$ and $R(\beta_i) < R(\beta_j)$ or $R(\alpha_i) < R(\alpha_j)$ and $R(\beta_i) > R(\beta_j)$, node pair (i, j) is discordant. If $R(\alpha_i) = R(\alpha_j)$ and/or $R(\beta_i) = R(\beta_j)$, node pair (i, j) is neither concordant nor discordant. It is defined as follows:

$$\tau_b(R(\alpha), R(\beta)) = \frac{n_c - n_d}{\sqrt{(n_c + n_{disc} + u)(N_c + N_d + v)}} \quad (11)$$

where n_c and n_d stand for the number of concordant and discordant pairs, respectively, and u and v hold the number of tied pairs in sets $R(\alpha)$ and $R(\beta)$, respectively.

Rank-Biased Overlap (RBO) [28] is capable of placing more emphasis on the top nodes between the two ranked lists $R(\alpha)$ and $R(\beta)$ of classical and community-aware centrality measures. Its value ranges between $[0, 1]$. It is defined as follows:

$$RBO(R(\alpha), R(\beta)) = (1 - p) \sum_{d=1}^{\infty} p^{(d-1)} \frac{|R(\alpha_d) \cap R(\beta_d)|}{d} \quad (12)$$

where p dictates “user persistence” and the weight to the top ranks, d is the depth reached on sets $R(\alpha)$ and $R(\beta)$, and $|R(\alpha_d) \cap R(\beta_d)|/d$ is the proportion of the similarity overlap at depth d . Note that p is set to 0.9 in the experiments.

Infomap Community Detection Algorithm [24] is based on compression of information. The idea is that a random walker on a network is likely to stay longer inside a given community and shorter outside communities. Accordingly, using Huffman coding, each community is defined by a unique codeword and nodes inside communities are defined by other codewords that can be reused in different communities. The optimization algorithm minimizes the coding resulted by the path of the random walker, achieving a concise map of the community structure.

4 Experimental Results

In this section, the results of the experiments performed on the real-world networks are reported. The first set of experiments involves calculating Kendall’s Tau correlation coefficient for all possible combinations between classical and community-aware centrality measures. The second experiment involves calculating the RBO similarity across all the combinations.

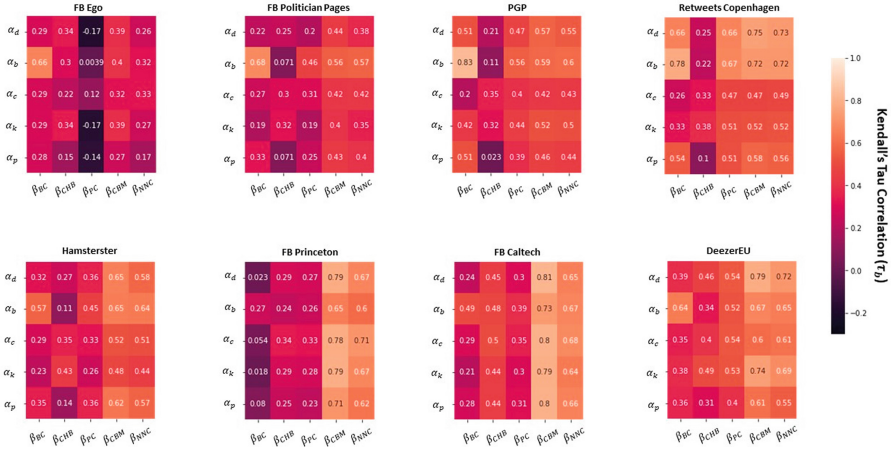


Fig. 1. Heatmaps of the Kendall’s Tau correlation (τ_b) of real-world networks across the various combinations between classical (α) and community-aware (β) centrality measures. The classical centrality measures are: $\alpha_d = \text{Degree}$, $\alpha_b = \text{Betweenness}$, $\alpha_c = \text{Closeness}$, $\alpha_k = \text{Katz}$, $\alpha_p = \text{PageRank}$. The community-aware centrality measures are: $\beta_{BC} = \text{Bridging centrality}$, $\beta_{CHB} = \text{Community Hub-Bridge}$, $\beta_{PC} = \text{Participation Coefficient}$, $\beta_{CBM} = \text{Community-based Mediator}$, $\beta_{NNC} = \text{Number of Neighboring Communities}$. (Color figure online)

4.1 Correlation Analysis

Kendall Tau’s correlation is applied on each network given all of the possible combinations between the 5 classical and 5 community-aware centrality measures. The 25 different combinations of the Kendall Tau’s correlation for the 8 OSNs are reported in Fig. 1. The Kendall’s Tau values range from -0.17 to 0.83 . Low correlation from -0.17 to 0.3 is characterized by the dark purple color of the heatmaps. Medium correlation from 0.3 to 0.6 is characterized by the fuchsia color. High correlation above 0.6 is characterized by the light pink color.

Networks’ heatmaps are arranged from low correlation (FB Ego) to medium-high (DeezerEU) correlation between classical and community-aware centrality measures. Heatmaps show that there are different behaviors among the community-aware centrality measures under study when they are compared to classical centrality measures. Specifically, Bridging centrality (β_{BC}), Community Hub-Bridge (β_{CHB}) and Participation Coefficient (β_{PC}) show consistency in their low correlation with classical centrality measures. On the other hand, Community-based Mediator (β_{CBM}) and Number of Neighboring Communities (β_{NNC}) vary across networks. In FB Ego, FB Politician Pages, and PGP, the correlation values are in the low to medium range, while in Hamsterster, FB Princeton, FB Caltech, and DeezerEU they are in the medium to high range.

Note that in Retweets Copenhagen network, the community-aware centrality measures show high correlation with the classical centrality measures degree and betweenness but low to medium correlation with the others. This is with the

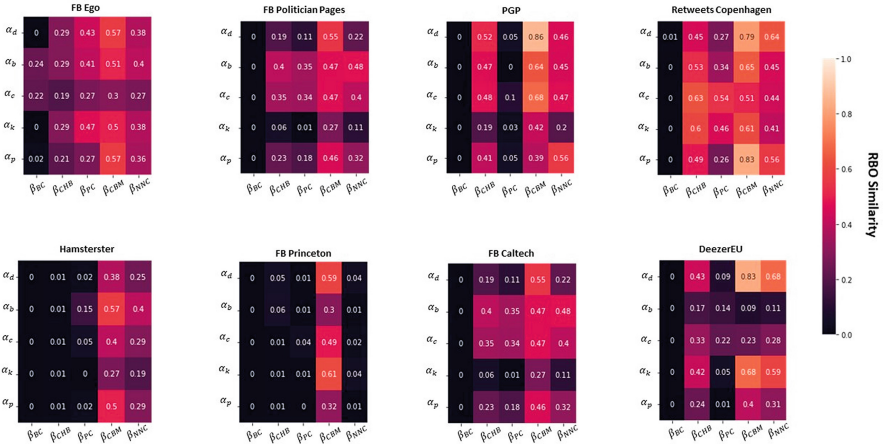


Fig. 2. Heatmaps of the RBO similarity at $p = 0.9$ of real-world networks across the various combinations between classical (α) and community-aware (β) centrality measures. The classical centrality measures are: α_d = Degree, α_b = Betweenness, α_c = Closeness, α_k = Katz, α_p = PageRank. The community-aware centrality measures are: β_{BC} = Bridging centrality, β_{CHB} = Community Hub-Bridge, β_{PC} = Participation Coefficient, β_{CBM} = Community-based Mediator, β_{NNC} = Number of Neighboring Communities. (Color figure online)

exception of Community Hub-Bridge (β_{CHB}) which shows low correlation with all classical centrality measures.

This experiment aims to answer the main research question, that is, do community-aware centrality measures provide distinctive information about the members within OSNs when compared to classical centrality measures? Results show that community-aware centrality measures indeed provide different information from that of classical centrality measures to the members within OSNs. Nonetheless, Bridging centrality (β_{BC}), Community Hub-Bridge (β_{CHB}), and Participation Coefficient (β_{PC}) show consistency in providing distinctive information to the members of 8 networks at hand. They always show low correlation. While Community-based Mediator (β_{CBM}) and Number of Neighboring Communities (β_{NNC}) show discrepancy in their behavior from one network to another.

4.2 Similarity Analysis

As top nodes are more important than bottom nodes in centrality assessment, RBO is calculated. Moreover, high correlation doesn't necessarily mean high similarity. This is more obvious when ties exist among the rankings of a set. Figure 2 shows the RBO similarity heatmaps of the 8 OSNs. The RBO values range from 0 to 0.86. Low similarity from 0 to 0.3 is characterized by the dark purple color. Medium similarity from 0.3 to 0.6 is characterized by the fuchsia color. High similarity over 0.6 is characterized by the light pink color. For comparison purposes, the networks are arranged in the same order as in Fig. 1.

Inspecting the heatmaps, Bridging centrality (β_{BC}) shows almost no similarity with all other classical centrality measures. To a less extent come Community Hub-Bridge (β_{CHB}) and Participation Coefficient (β_{PC}) community-aware centrality measures. For these community-aware centralities, the low similarity is consistent across the networks. Community-based Mediator (β_{CBM}) and Number of Neighboring Communities (β_{NNC}) change from one network to another. For example, taking the RBO similarity of the combination (α_d, β_{NNC}) in DeezerEU, it is equal to 0.68 while in FB Princeton it is equal to 0.04.

This experiment shows consistency with the previous experiment. Indeed, Bridging centrality (β_{BC}), Community Hub-Bridge (β_{CHB}), and Participation Coefficient (β_{PC}) community-aware centrality measures show the lowest similarity to classical centrality measures and their behavior is consistent across the 8 OSNs under study. This case is similar to the case under Kendall Tau's correlation. However, RBO is more extreme than Kendall's Tau correlation, where low values of similarity can be seen. This is simply due to the RBO definition accounting for ranks. When a group of nodes acquires the same rank, as RBO moves from depth d to $d+1$, the group of tied nodes occurring at d are surpassed and hence account less to the similarity between the two ranked lists.

Referring back to the main research question, indeed, community-aware and classical centrality measures do not convey the same information. Nonetheless, these measures can be divided into two groups. The first group has consistent low similarity with the classical centrality measures while the second group has varying similarity across the networks.

5 Conclusion

Communities have major consequences on the dynamics of a network. Humans tend to form communities within their social presence according to one or many similarity criteria. In addition to that, humans tend to follow other members manifesting power, influence, or popularity, resulting in dense community structures. Centrality measures aim to identify the key members within OSNs, which is crucial for a lot of strategic applications. However, these measures are agnostic to the community structure. Newly developed centrality measures account for the existence of communities.

Most works have been conducted on classical centrality measures on online social networks. In this work, we shed the light on the relationship between classical and community-aware centrality measures in OSNs. Using 8 real-world OSNs from different platforms, their community structure is uncovered using Infomap. Then, for each network, 5 classical and 5 community-aware centrality measures are calculated. After that, correlation and similarity evaluation between all possible classical and community-aware centrality measures is conducted. Results show that globally these two types of centrality do not convey the same information. Moreover, community-aware centrality measures exhibit two behaviors. The first set (Bridging centrality, Community Hub-Bridge, and Participation Coefficient) exhibit low correlation and low similarity for all the networks under

study. The second set (Community-based Mediator and Number of Neighboring Communities) shows varying correlation and similarity across networks.

Results of this study suggest that community-aware centrality measures are worth looking into when searching for key members in OSNs, as they provide different information from classical centrality measures. This work opens future research directions. Further study will investigate the effect of network topology on the relationship between classical and community-aware centrality measures and whether results are consistent using different community detection algorithms.

References

1. Jalili, M., Perc, M.: Information cascades in complex networks. *J. Complex Netw.* **5**(5), 665–693 (2017)
2. Wang, Z., Moreno, Y., Boccaletti, S., Perc, M.: Vaccination and epidemics in networked populations—an introduction (2017)
3. Azzimonti, M., Fernandes, M.: Social media networks, fake news, and polarization. Technical report, National Bureau of Economic Research (2018)
4. Lü, L., Chen, D., Ren, X.-L., Zhang, Q.-M., Zhang, Y.-C., Zhou, T.: Vital nodes identification in complex networks. *Phys. Rep.* **650**, 1–63 (2016)
5. Sciarra, C., Chiarotti, G., Laio, F., Ridolfi, L.: A change of perspective in network centrality. *Sci. Rep.* **8**(1), 1–9 (2018)
6. Ibnoulouafi, A., El Haziti, M., Cherifi, H.: M-centrality: identifying key nodes based on global position and local degree variation. *J. Stat. Mech: Theory Exp.* **2018**(7), 073407 (2018)
7. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* **99**(12), 7821–7826 (2002)
8. Jebabli, M., Cherifi, H., Cherifi, C., Hamouda, A.: User and group networks on Youtube: a comparative analysis. In: 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), pp. 1–8. IEEE (2015)
9. Cherifi, H., Palla, G., Szymanski, B.K., Lu, X.: On community structure in complex networks: challenges and opportunities. *Appl. Netw. Sci.* **4**(1), 1–35 (2019)
10. Hwang, W., Cho, Y., Zhang, A., Ramanathan, M.: Bridging centrality: identifying bridging nodes in scale-free networks. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 20–23 (2006)
11. Ghalmane, Z., El Hassouni, M., Cherifi, H.: Immunization of networks with non-overlapping community structure. *Soc. Netw. Anal. Min.* **9**(1), 45 (2019)
12. Guimera, R., Amaral, L.A.N.: Functional cartography of complex metabolic networks. *Nature* **433**(7028), 895–900 (2005)
13. Tulu, M.M., Hou, R., Younas, T.: Identifying influential nodes based on community structure to speed up the dissemination of information in complex network. *IEEE Access* **6**, 7390–7401 (2018)
14. Gupta, N., Singh, A., Cherifi, H.: Community-based immunization strategies for epidemic control. In: 2015 7th International Conference on Communication Systems and Networks (COMSNETS), pp. 1–6. IEEE (2015)
15. Chakraborty, D., Singh, A., Cherifi, H.: Immunization strategies based on the overlapping nodes in networks with community structure. In: International Conference on Computational Social Networks, pp. 62–73. Springer, Cham (2016)

16. Kumar, M., Singh, A., Cherifi, H.: An efficient immunization strategy using overlapping nodes and its neighborhoods. In: Companion Proceedings of the The Web Conference 2018, pp. 1269–1275 (2018)
17. Ghalmane, Z., Cherifi, C., Cherifi, H., El Hassouni, M.: Centrality in complex networks with overlapping community structure. *Sci. Rep.* **9**(1), 1–29 (2019)
18. Li, C., Li, Q., Van Mieghem, P., Stanley, H.E., Wang, H.: Correlation between centrality metrics and their application to the opinion model. *Eur. Phys. J. B* **88**(3), 1–13 (2015)
19. Oldham, S., Fulcher, B., Parkes, L., Arnatkevičiūtė, A., Suo, C., Fornito, A.: Consistency and differences between centrality measures across distinct classes of networks. *PLoS One* **14**(7) (2019)
20. Shao, C., Cui, P., Xun, P., Peng, Y., Jiang, X.: Rank correlation between centrality metrics in complex networks: an empirical study. *Open Phys.* **16**(1), 1009–1023 (2018)
21. Landherr, A., Friedl, B., Heidemann, J.: A critical review of centrality measures in social networks. *Bus. Inf. Syst. Eng.* **2**, 371–385 (2010)
22. Grando, F., Noble, D., Lamb, L.C.: An analysis of centrality measures for complex and social networks. In: 2016 IEEE Global Communications Conference (GLOBECOM), pp. 1–6. IEEE (2016)
23. Rajeh, S., Savonnet, M., Leclercq, E., Cherifi, H.: Interplay between hierarchy and centrality in complex networks. *IEEE Access* **8**, 129717–129742 (2020)
24. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci.* **105**(4), 1118–1123 (2008)
25. Rossi, R., Ahmed, N.: The network data repository with interactive graph analytics and visualization. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
26. Rozemberczki, B., Sarkar, R.: Characteristic functions on graphs: birds of a feather, from statistical descriptors to parametric models (2020)
27. Kunegis, J.: Handbook of network analysis [konect—the koblenz network collection]. [arXiv:1402.5500](https://arxiv.org/abs/1402.5500) (2014). <http://konect.cc/networks/>
28. Webber, W., Moffat, A., Zobel, J.: A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst. (TOIS)* **28**(4), 1–38 (2010)

Network Analysis



Complex Network Analysis of North American Institutions of Higher Education on Twitter

Dmitry Zinoviev^(✉), Shana Cote, and Robert Díaz

Suffolk University, Boston, MA 02114, USA
dzinoviev@suffolk.edu, {scote4,rdiaz2}@su.suffolk.edu

Abstract. North American institutions of higher education (IHEs): universities, 4- and 2-year colleges, and trade schools—are heavily present and followed on Twitter. An IHE Twitter account, on average, has 20,000 subscribers. Many of them follow more than one IHE, making it possible to construct an IHE network, based on the number of co-followers. In this paper, we explore the structure of a network of 1,435 IHEs on Twitter. We discovered significant correlations between the network attributes: various centralities and clustering coefficients—and IHEs’ attributes, such as enrollment, tuition, and religious/racial/gender affiliations. We uncovered the community structure of the network linked to homophily—such that similar followers follow similar colleges. Additionally, we analyzed the followers’ self-descriptions and identified twelve overlapping topics that can be traced to the followers’ group identities.

Keywords: Complex networks · Higher education · Computational social science

1 Introduction

According to the National Center for Education Statistics [6], in 2018, there were 4,313 degree-granting postsecondary institutions, also known as institutions of higher education (IHEs), in the USA. This number includes public and private (both nonprofit and for-profit) universities, liberal arts colleges, community colleges, religious schools, and trade schools.

The IHEs enjoy a heavy presence on social media, in particular, on Twitter. In 2012, Linvill *et al.* [3] found that IHEs employ Twitter primarily as an institutional news feed to a general audience. These results were confirmed by Kimmons *et al.* in 2016 [2] and 2017 [14]; the authors further argue that Twitter failed to become a “vehicle for institutions to extend their reach and further demonstrate their value to society”—and a somewhat “missed opportunity for presidents to use Twitter to connect more closely with alumni and donors” [15]. The same disconnect has been observed for IHE library accounts [11].

Despite the failed promise, the IHEs massively invest in online marketing [12] and, in reciprocity, collect impressive follower lists that include both organizations and individuals. The longer follower lists demonstrate a positive effect on

IHE performance, particularly, on student recruitment [9], and may eventually affect IHE ratings or at least correlate with them [4]. Therefore, follower lists are essential marketing instruments and should be studied comprehensively.

To the best of our knowledge, this paper is the first attempt to look at a social network of IHE Twitter accounts based on the similarities of their follower lists. We hypothesize that the exogenous parameters, such as enrollment, tuition, and religious/gender/race preferences, affect the structure of the network and positions/importance of the IHEs in it.

The rest of the paper is organized as follows: In Sect. 2, we describe the data set, its provenance, and structure; in Sect. 3, we explain the network construction; in Sect. 4, we go over the network analysis, and present the results; in Sect. 5, we take a look at the followers; in Sect. 6, we discuss the results. Finally, in Sect. 7, we conclude.

2 Data Set

Our data set consists of two subsets: social networking data from Twitter and IHE demographics from Niche [8]. We used the former to construct a network of IHEs and the latter to provide independent variables for the network analysis. Both subsets were collected in Summer 2020.

The Twitter data set describes the Twitter accounts of 1,450 IHEs from all 50 states and the District of Columbia. The majority of the accounts are the official IHE accounts, but for some IHEs, we had to rely on secondary accounts, such as those of admission offices or varsity sports teams. For each IHE, we have the following attributes (and their mean values): geographical location (including the state), the lists of followers (20,198) and friends (1,130), the numbers of favorites (“likes”; 4,656) and statuses (“posts”; 9,132), the account age in years (10.4), and whether the account is verified or not (32% accounts are verified).

With some IHEs having more than a million followers (e.g., MIT and Harvard University), we chose to restrict our lists to up to 10,000 followers per IHE. This limitation may have resulted in a slight underestimation of the connectedness of the most popular IHEs. We explain in Subsect. 3.1 why we believe that the underestimation is not crucial.

It is worth noting that while we have downloaded the friend lists, we do not use them in this work because they are controlled by the IHE administrations/PR offices and cannot be considered truly exogenous.

The combined list of followers consists of 347,920 users. This number does not include the “occasional” followers who subscribed to fewer than three IHEs.

The descriptive IHE data comes from Niche [8], an American company that provides demographics, rankings, report cards, and colleges’ reviews. It covers 1,435 of the IHEs that we selected for the network construction. Five more IHEs were not found on Niche and, though included in the network, were not used in further analysis.

For each IHE, we have the following attributes:

Binary:

- “Liberal Arts” college designation,
- Application options: “SAT/ACT Optional”, “Common App Accepted,” or “No App Fee” (these options can be combined).

Categorical:

- Type: “Private”, “Public”, “Community College”, or “Trade School”; note that all community colleges and trade schools in our data set are public;
- Religious affiliation: “Christian”, “Catholic”, “Muslim” or “Jewish”; we lumped the former two together;
- Online learning options: “Fully Online”, “Large Online Program”, or “Some Online Degrees”;
- Gender preferences: “All-Women” or “All-Men”;
- Race preferences: “Hispanic-Serving Institution” (HSI) or “Historically Black College or University” (HBCU).

Count or real-valued: Enrollment and tuition. We noticed that due to the broad range of enrollments and tuition, enrollment and tuition logarithms are better predictors. We will use $\log(\text{enrollment})$ and $\log(\text{tuition})$ instead of enrollment and tuition throughout the paper.

3 Network Construction

We define the network G of IHEs on Twitter as $G = (N, E)$. Here, $N = \{n_i\}$ is a set of 1,450 nodes, each representing an IHE account, and $E = \{e_{ij}\}$ is a set of weighted edges.

Let $f(n)$ be a set of followers of the account n . As noted in Sect. 2, $\forall n \in N : \#f(n) \leq 10,000$.

Let $f^{-1}(q) = \{n \in N \mid q \in f(n)\}$ be a set of all IHE accounts followed by user q . Note that q itself may be a member of N : IHEs can follow each other.

The definition of an edge is derived from the concept of G as a network based on co-following: two nodes n_i and n_j share an edge e_{ij} iff they have at least one shared follower that also follows at least three IHE accounts. We denote a set of such qualified followers as Q :

$$Q = \{q \mid \#f^{-1}(q) \geq 3\} \quad (1)$$

$$\forall i, j : \exists e_{ij} \Leftrightarrow Q \cap f(n_i) \cap f(n_j) \neq \emptyset \quad (2)$$

The number of edges in G is, therefore, 928,476. The network is connected (there is only one connected component) and quite dense: its density is 0.88.

Finally, let $w_{ij} > 0$ be the weight of the edge e_{ij} . We initially define w_{ij} as the number of qualified shared followers:

$$w_{ij} = \#(Q \cap f(n_i) \cap f(n_j)). \quad (3)$$

The choice of the number of shared followers as the edge weight—rather than, say, the Jaccard similarity suggested by a friendly reviewer—was dictated by the long-tail nature of the distribution of the number of followers. If Jaccard similarity were used, an IHE A with many followers would never be similar to any IHE B with few followers, even if all B 's followers also follow A . We essentially postulate that anyone following two IHEs makes them more similar than anyone following just one of them makes them dissimilar.

The resulting weights are large (on the order of 10^3 – 10^4), while many network algorithms, such as community detection and visualization, expect them to be in the range $(0 \dots 1]$. We used the algorithm proposed in [10] to normalize the weights without affecting the calculated node attributes.

3.1 A Note on Edge Weight Calculations

We mentioned in Sect. 2 that we use only up to 10,000 followers for edge weight calculations. The truncated follower lists result in lower weights. We can estimate the difference between true and calculated weights by assuming the worst-case scenario: The shared followers are uniformly distributed in the follower lists. Let $F = \overline{\#f} = 21,123$ be the mean number of followers; let $T = 10,000$; let $p \approx 0.685$ be the probability that a follower list is not longer than T ; let \overline{w} be the mean edge weight; finally, let $\overline{w^*}$ be the estimated mean edge weight. Note that if $p = 1$ then $\overline{w^*} = \overline{w}$. One can show that:

$$\frac{\overline{w^*}}{\overline{w}} \approx \left(\frac{(F - T)p + T}{F} \right)^2 \approx 1.436. \quad (4)$$

Seemingly, the weights of all edges that are incident to at least one node with a truncated follower list are underestimated by $\approx 30\%$.

However, we noticed that Twitter reports follower lists not uniformly but roughly in the order of prominence: the prominent followers with many followers of their own are reported first. We hope that the shared users responsible for edge formations are mostly reported among the first 10,000 followers.

4 Network Analysis

In this section, we analyze the constructed network and present the results. We looked at individual nodes' positions in the network (monadic analysis), relations between adjacent nodes (dyadic analysis), and node clusters (community analysis).

4.1 Monadic Analysis

We used Python library *networkx* [17] to calculate the monadic attributes: degree, closeness, betweenness, and eigenvector centralities, and local clustering coefficient—for each node $n \in G$. All the centralities of n express various

aspects of n 's prominence in a network [16]: the number of closely similar IHEs (degree), the average similarity of n to all other IHEs (closeness), the number of IHEs that are similar to each other by being similar to n (betweenness), and the measure of mutual importance (eigenvector: " n is important if it is similar to other important nodes"). The local clustering coefficient reports if the nodes similar to n are also similar to each other.

We use multiple ordinary least squares (OLS) regression to model the relationships between each of the network attributes and the following independent variables: tuition, enrollment, Twitter account age, Twitter account verified status, "No App Fee", "Liberal Arts" designation, "SAT/ACT Optional", "Common App Accepted", race preferences, online learning options, type/religious affiliations, and gender preferences (see Sect. 2). We combined the IHE type and religious affiliations into one variable because all public schools are secular.

The number of samples in the regression is 1,348 (the intersection of the Niche set and Twitter set). Table 1 shows the independent variables that significantly ($p \leq 0.01$) explain the monadic network measures, and the regression coefficients.

Table 1. Variables that significantly ($p \leq 0.01$) explain the monadic network measures: betw[enness], clos[enness], degr[ee], eigen[vector] centralities, clust[ering] coefficient, and numbers of favorites ("likes"), followers, friends, and statuses (posts). †The marked rows represent levels of the categorical variables.

Variable	Coef.								
	Betw.	Clos.	Clust.	Degr.	Eigen.	Favorites	Followers	Friends	Posts
Liberal Arts	0.27	0.05		0.08	0.08		-0.98		
Private†	0.29						1.40		
Account Age	0.12	0.02		0.03	0.03				0.05
Tuition	-0.21								
Common App		0.02							
No App Fee		0.03		0.05	0.05				
Large Online†		0.05		0.09	0.09				
Some Online†		0.02		0.04	0.04		-0.55		
HBCU†		0.06		0.11	0.10				
Christian†	-0.04		0.07	0.07					0.63
Verified		-0.03		-0.05	-0.05	0.66	1.30	0.52	0.48
Enrollment		0.03	0.01	0.06	0.06	0.33	0.65	0.29	0.29

4.2 Dyadic Analysis

The only dyadic variable in our model is the edge weight. As a reminder, the weight of an edge is derived from the number of Twitter co-followers of the incident nodes. A stronger edge indicates a larger overlap of the followers and, presumably, a closer similarity between the IHEs, even if the nature and reason for the similarity is unclear.

We hypothesize that, because of homophily, edge weights depend on the difference between the incident node attributes. We calculate the dyadic versions of the monadic independent variables for the OLS regression modeling as follows:

For the binary and categorical variables: A calculated dyadic variable y equals 1 if the values of the underlying monadic variable x differ, and 0, otherwise:

$$y_{ij} = \begin{cases} 0 & \text{if } x_i = x_j \\ 1 & \text{if } x_i \neq x_j \end{cases} \quad (5)$$

For example, if both incident nodes represent liberal art colleges, then the dyadic “Same Liberal Arts designation” variable for the edge is 0.

For the count or real-valued variables: A calculated dyadic variable y equals the absolute value of the arithmetic difference of the underlying monadic x variable at the incident nodes:

$$y_{ij} = |x_i - x_j|. \quad (6)$$

Both clauses emphasize the difference of the monadic attributes along the incident edge. Table 2 shows the independent variables that significantly ($p \leq 0.01$) explain the edge weights, and the regression coefficients. For this analysis, we add the state in which an IHE is located to the monadic variables listed in Subsect. 4.1.

Table 2. Variables that significantly ($p \leq 0.01$) explain the edge weights

Variable	Coef.
Same state	0.0169
Similar enrollment	0.0024
Similar tuition	0.0022
Same religious affiliation	0.0019
Same online preferences	0.0010
Similar account age	0.0008
Same “Common App Accepted” option	0.0008
Same “No App Fee” option	0.0008
Same race designation	0.0006
Same “SAT/ACT Optional” option	0.0005
Both verified	-0.0001
Same gender designation	-0.0022
Same “Liberal Arts” designation	-0.0026

4.3 Community Analysis

We used the Louvain community detection algorithm [1] to partition G into network communities, or clusters: tightly connected non-overlapping groups of

nodes with more internal connections than external connections. We requested a resolution of 0.8 (lower than the standard 1.0) to discover smaller clusters and, as a result, partitioned G into 22 disjoint clusters $C = \{c_i\}$. The Newmann modularity [7] of the partition is 0.152 on the scale $[-1/2 \dots 1]$. Each cluster contains the nodes representing the IHEs that are somewhat more similar to each other than to an IHE from another cluster. In other words, the level of homophily within a cluster is higher than between the clusters. We expect to identify the independent variables responsible for the homophily.

Table 3 shows the independent variables that significantly ($p \leq 0.01$) explain the membership in select clusters, and the regression coefficients. Note that the clusters 6, 9, 10, 16, and 19 do not have any significant explanatory variables, and the clusters 18, 20, 21, and 22 are single-node isolates.

Table 3. Variables that significantly ($p \leq 0.01$) explain membership in select clusters. (See Fig. 1.) †The marked rows represent levels of the categorical variables.

Variable	Coef.													
	1	2	3	4	5	7	8	11	12	13	14	15	17	
Christian†			1.75				-3.85							
Comm. Coll.†	2.75													
Common App			-1.95				2.38		1.75					
Enrollment			-0.53	1.99	-0.50		-0.73			-0.56				
HBCU†						7.29								
HSI†								1.30			1.65			
Large Online†														3.20
Liberal Arts							-1.77						2.97	
No App Fee	1.19													
Private†							-2.88							
SAT/ACT Opt	1.44		-0.74											-2.08
Some Online†					0.92				-1.61					
Trade School†	3.34	-2.42												
Tuition	-1.85				1.68		3.14		-1.05					
Verified	-1.34		-1.27	1.80					-1.27					

As a side note, community detection can be used to visualize G . Large networks are usually hard to visualize, especially when their Newmann modularity is low, and the community structure is not prominent. We use the extracted partition C to build a bird’s-eye view of G , known as an induced network $I = (C, E^I)$ (Fig. 1). An induced node in I represents a cluster in G . An induced edge between two nodes c_i and c_j in I exists iff there exists at least one edge from any node in c_i to any node in c_j :

$$\forall i, j : \exists e_{ij}^I \Leftrightarrow (\exists k, l : n_k \in c_i \wedge n_l \in c_j \wedge \exists e_{kl}). \tag{7}$$

Respectively, the weight of such induced edge w_{ij}^I is the number of the original edges in G from any node in c_i to any node in c_j :

$$w_{ij}^I = \#\{e_{kl} \mid n_k \in c_i \wedge n_l \in c_j\}. \tag{8}$$

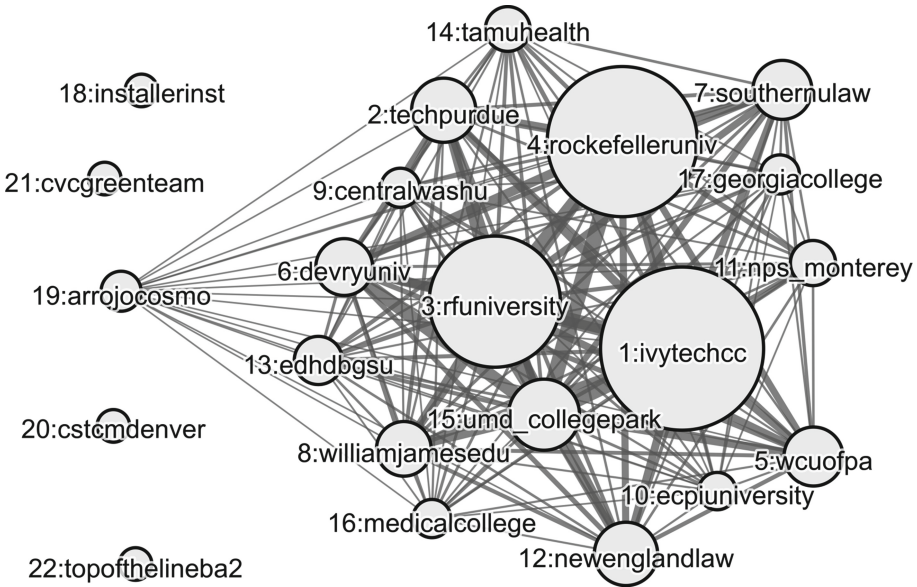


Fig. 1. An induced network of IHE clusters. Each node represents a cluster named after its highest-enrollment IHE. The node size represents the number of IHEs in the cluster. The edge width represents the number of IHE-level connections.

The name of each cluster in Fig. 1 incorporates the name of the Twitter account of the IHE with the highest enrollment in the cluster.

5 Followers' Analysis

At the last stage of the network analysis, we shift the focus of attention from the IHEs to their followers.

We selected 14,750 top followers who follow at least 1% of the IHEs in our data set. Approximately 8% of them have an empty description or a description in a language other than English. Another 268 accounts belong to the IHEs from the original data set, and at least 326 more accounts belong to other IHEs, both domestic and international.

We constructed a semantic network of lemmatized tokens by connecting the tokens that frequently (10 or more times) occur together in the descriptions. We applied the Louvain [1] community detection algorithm to extract topics—the clusters of words that are frequently used together. The algorithm identified twelve topics named after the first nine most frequently used words. For each follower's account, we selected the most closely matching topics. The names and counts for the most prominent topics are shown in Table 4.

Even after the manual cleanup, some of the 12,984 remaining followers' accounts probably still belong to IHEs and associated divisions, organizations,

Table 4. The most prominent topics and the number of followers accounts that use them. (Since a description may contain words from more than one topic, the sum of the counts is larger than the number of followers.) [†]Topic #8 is technical.

ID	Top seven topic terms	Count
1	Education, service, higher, business, professional, solution, research	5,039
2	Student, program, online, academic, year, helping, opportunity	3,460
3	School, high, official, twitter, news, account, follow	3,308
4	College, community, campus, institution, mission, member, black	2,870
5	Help, life, world, love, social, work, people	2,258
6	University, career, state, new, job, find, best	1,814
7	Coach, teacher, author, husband, father, writer, book	1,125
8 [†]	Endorsement, like, link, facebook, retweets, equal, following	557
9	Lover, mom, wife, mother, dog	515

and officials. This deficiency would explain the significance of the topics #4 and, partially, #2 that seem to use the endogenous terminology. The remaining topics are exogenous to the IHEs and represent higher education services, high schools, communities, career services, and individuals (“male” and “female”).

6 Discussion

Based on the results from Sect. 4, we look at each independent variable’s influence on each network and Twitter performance parameter, whenever the influence is statistically significant ($p \leq 0.01$).

It has been observed [13] that the centrality measures are often positively correlated. Indeed, in G ’s case, we saw strong (≥ 0.97) correlations between the degree, eigenvector, and closeness centralities, which explains their statistically significant connection to the same independent variables (Table 1). More central nodes tend to represent:

Some specialty IHEs: Liberal arts colleges, HBCUs.

Internet-savvy IHEs: IHEs with a longer presence on Twitter, IHEs with some or many online programs.

Bigger IHEs with simplified application options: IHEs with no application fees (and accepting Common App—for the closeness centrality), larger IHEs.

All these IHEs blend better in their possibly non-homogeneous network neighborhoods.

The betweenness centrality—the propensity to act as a shared reference point—is positively affected by being a liberal arts college or private IHE, and

longer presence on Twitter, and negatively affected by higher tuition and being a Christian IHE. On the contrary, large and Christian IHEs tend to have a larger local clustering coefficient and a more homogeneous network neighborhood.

All Twitter performance measures: the numbers of favorites, followers, friends, and posts—are positively affected by enrollment and the verified account status. The number of posts is also higher for the IHEs with a more prolonged presence on Twitter and Christian IHEs. The number of followers is also higher for private IHEs and lower for liberal arts colleges and IHEs with some online programs. The latter observation is counterintuitive and needs further exploration.

Edge weight is the only dyadic variable in G . Table 2 shows that the weight of an edge is explained by the differences of the adjacent nodes' attributes. Some of the attributes promote homophily, while others inhibit it.

The strongest edges connect the IHEs located in the same state, which is probably because many local IHEs admit the bulk of the local high schools' graduates and are followed by them and their parents. Much weaker, but still positive, contributors to the edge weight are similar enrollment and tuition, same religious affiliation, online teaching preferences, racial preferences, and application preferences, a “classical” list of characteristics that breed connections [5]. We hypothesize that prospective students and their parents follow several IHEs that match the same socio-economic profile. National, regional, and professional associations (such as the National Association for Equal Opportunity and National Association of Independent Colleges and Universities) may follow similar IHEs for the same reason.

We identified two factors that have a detrimental effect on edge weight: having the same gender designation (“All-Male”, “All-Female”, or neither) and especially the same “Liberal Arts” designation. There are 1.58% of “All-Female” IHEs (and no “All-Male”) and 11.2% Liberal Arts colleges in our data set. The IHEs of both types may be considered unique and not substitutable, thus having fewer shared followers.

In the same spirit, some network communities (clusters) of G represent compact groups of IHEs with unique characteristics (Table 3). For example, cluster 1 tends to include community colleges and trade schools with no application fees, optional SAT/ACT, and lower tuition (e.g., Carl Sandburg College). Cluster 3 is a preferred locus of smaller Christian IHEs that do not accept Common App but require SAT/ACT (New Saint Andrews College). The last comprehensive example is cluster 8: smaller public, secular, expensive IHEs embracing Common App (University of Maine at Machias). IHEs with large online programs are in cluster 17 (Middle Georgia State University), Historically Black Colleges and Universities—in cluster 7 (North Carolina A&T State University), and Liberal Arts colleges—in cluster 15 (St. Olaf College).

It is worth reiterating that the membership in five clusters containing 9.1% IHEs, cannot be statistically significantly explained by any independent variable. The explanatory variables, if they exist, must be missing from our data set.

7 Conclusion

We constructed and analyzed a social network of select North American institutions of higher education (IHEs) on Twitter, using the numbers of shared followers as a measure of connectivity. We used multiple OLS regression to explain the network characteristics: centralities, clustering coefficients, and cluster membership. The regression variables include IHE size, tuition, geographic location, type, and application preferences. We discovered statistically significant connections between the independent variables and the network characteristics. In particular, we observed strong homophily among the IHEs in terms of the number of shared followers. Finally, we analyzed the self-provided descriptions of the followers and assigned them to several classes. Our findings may help understand the college application decision-making process from the points of view of the major stakeholders: applicants, their families, high schools, and marketing and recruitment companies.

References

1. Blondel, V., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech.: Theory Exp.* **2008**(10), 1000 (2008)
2. Kimmons, R., Veletsianos, G., Woodward, S.: Institutional uses of Twitter in U.S. higher education. *Innov. High. Educ.* **42**, 97–111 (2017)
3. Linvill, D., McGee, S., Hicks, L.: Colleges' and universities' use of Twitter: a content analysis. *Public Relat. Rev.* **38**(4), 636–638 (2012)
4. McCoy, C., Nelson, M., Weigle, M.: University Twitter engagement: using Twitter followers to rank universities. arXiv preprint [arXiv:1708.05790](https://arxiv.org/abs/1708.05790) (2017)
5. McPherson, M., Smith-Lovin, L., Cook, J.: Birds of a feather: homophily in social networks. *Ann. Rev. Sociol.* **27**(1), 415–444 (2001)
6. National Center for Education Statistics: Degree-granting postsecondary institutions, by control and level of institution (2018). https://nces.ed.gov/programs/digest/d18/tables/dt18_317.10.asp?current=yes. Accessed September 2020
7. Newman, M.: Modularity and community structure in networks. *Proc. Natl. Acad. Sci. U.S.A.* **103**(23), 8577–8696 (2006)
8. Niche: 2020 Best Colleges in America. <https://niche.com/colleges>. Accessed 2020
9. Rutter, R., Roper, S., Lettice, F.: Social media interaction, the university brand and recruitment performance. *J. Bus. Res.* **69**(8), 3096–3104 (2016)
10. Simas, T., Rocha, L.M.: Distance closures on complex networks. *Netw. Sci.* **3**(2), 227–268 (2015)
11. Stewart, B., Walker, J.: Build it and they will come? Patron engagement via Twitter at historically black college and university libraries. *J. Acad. Libr.* **44**(1), 118–124 (2018)
12. Taylor, Z., Bicak, I.: Buying search, buying students: how elite U.S. institutions employ paid search to practice academic capitalism online. *J. Mark. High. Educ.* **30**(2), 271–296 (2020)
13. Valente, T., Coronges, K., Lakon, C., Costenbader, E.: How correlated are network centrality measures? *Connections (Toronto Ont.)* **28**(1), 16 (2008)
14. Veletsianos, G., Kimmons, R., Shaw, A., Pasquini, L., Woodward, S.: Selective openness, branding, broadcasting, and promotion: Twitter use in Canada's public universities. *Educ. Media Int.* **54**(1), 1–19 (2017)

15. Walton, S.: Competing by tweeting: a content analysis of university presidents' tweets. Ph.D. thesis, University of North Dakota (2020)
16. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications, vol. 8. Cambridge University Press, Cambridge (1994)
17. Zinoviev, D.: Complex Network Analysis in Python. Pragmatic Bookshelf (2018)



Connectivity-Based Spectral Sampling for Big Complex Network Visualization

Jingming Hu¹, Seok-Hee Hong¹(✉), Jialu Chen¹, Marnijati Torkel¹, Peter Eades¹, and Kwan-Liu Ma²

¹ The University of Sydney, Sydney, Australia
{jihu2855, seokhee.hong, jche6589, mtor0581, peter.eades}@sydney.edu.au

² University of California at Davis, Davis, USA
ma@cs.ucdavis.edu

Abstract. Graph sampling methods have been used to reduce the size and complexity of big complex networks for graph mining and visualization. However, existing graph sampling methods often fail to preserve the connectivity and important structures of the original graph.

This paper introduces a new divide and conquer approach to spectral graph sampling based on the graph connectivity (i.e., decomposition of a connected graph into biconnected components) and spectral sparsification. Specifically, we present two methods, spectral vertex sampling and spectral edge sampling by computing effective resistance values of vertices and edges for each connected component. Experimental results demonstrate that our new connectivity-based spectral sampling approach is significantly faster than previous methods, while preserving the same sampling quality.

1 Introduction

Big complex networks are abundant in many application domains, such as social networks and systems biology. Examples include facebook networks, protein-protein interaction networks, biochemical pathways and web graphs. However, good visualization of big complex networks is challenging due to scalability and complexity. For example, visualizations of big complex networks often produce hairball-like visualization, which makes it difficult for human to understand the structure of the graphs.

Graph sampling methods have been widely used to reduce the size of graphs in graph mining [6, 7]. Popular graph sampling methods include Random Vertex sampling, Random Edge sampling and Random Walk. However, previous work based on random sampling methods often fails to preserve the connectivity and important structures of the original graph, in particular for visualization [13].

Spectral sparsification is a technique to reduce the number of edges in a graph while retaining its structural properties [12]. More specifically, it is a stochastic sampling method, using the *effective resistance* values of edges, which is closely

Supported by the ARC Discovery Projects.

related to the commute distance of graphs. However, computing effective resistance values of edges is rather complicated, which can be very slow for big graphs [2].

This paper introduces a divide and conquer algorithm for spectral sparsification, based on the graph *connectivity*, called the BC (Block Cut-vertex) tree decomposition, which represents the decomposition of a graph into biconnected components. More specifically, the main idea is to divide a big complex network into biconnected components, and then compute the spectral sparsification for each biconnected component in parallel to reduce the runtime as well as to maintain the graph connectivity. Namely, the effective resistance values of edges are computed for each biconnected component, as an approximation of the effective resistance values of the original graph.

The main contribution of this paper is summarized as follows:

1. We present two new variations of spectral sparsification based on the connectivity, *spectral edge sampling* (BC_SS) and *spectral vertex sampling* (BC_SV). Note that the spectral edge sampling mainly sparsifies the edge set, however the spectral vertex sampling focuses on reducing the size of the vertex set.
2. Experimental results demonstrate that our BC_SS and BC_SV methods are significantly faster than the original SS (Spectral Sparsification) [2] and SV (Spectral Vertex sampling) [5], while preserving the same sampling quality, using comparison of the effective resistance values and rankings of edges (resp., vertices), sampling quality metrics, graph similarity, and visual comparison.

2 Related Work

2.1 Graph Sampling and Spectral Sparsification

Graph sampling methods have been extensively studied in graph mining to reduce the size of big complex graphs. Consequently, many stochastic sampling methods are available [6, 7]. For example, most popular stochastic sampling include Random Vertex sampling and Random Edge sampling. However, it was shown that random sampling methods often fail to preserve connectivity and important structure in the original graph, in particular for visualization [13].

Spielman et al. [12] introduced the *Spectral Sparsification (SS)*, a subgraph which preserves the structural properties of the original graph, and proved that every n -vertex graph has a spectral approximation with $O(n \log n)$ edges. More specifically, they presented a stochastic sampling method, using the *effective resistance* values of edges, which is closely related to the commute distance of graphs [12]. However, computing effective resistance values of edges is quite complicated and can be very slow for big graphs [2].

2.2 BC (Block Cut-Vertex) Tree Decomposition

The *BC tree* represents the tree decomposition of a connected graph G into biconnected components, which can be computed in linear time. There are two

types of nodes in the BC tree T ; a cut vertex c and a biconnected component B . A *cut vertex* is a vertex whose removal from the graph makes the resulting graph disconnected. A *biconnected component* (or *block*) is a maximal biconnected sub-graph.

2.3 Graph Sampling Quality Metrics

There are a number of quality metrics for graph sampling [6]. For our experiment, we use the following most popular quality metrics:

- *Degree Correlation Associativity (Degree)*: a basic structural metric, which computes the likelihood that vertices link to other vertices of similar degree, called positive degree correlation [9].
- *Closeness Centrality (Closeness)*: a centrality measure of a vertex in a graph, which sums the length of all shortest paths between the vertex and all the other vertices in the graph [3].
- *Average Neighbor Degree (AND)*: the measure of the average degree of the neighbors of each vertex [1].
- *Clustering Coefficient (CC)*: measures the degree of vertices which tend to cluster together [11].

3 BC Tree-Based Spectral Graph Sampling

We introduce a new divide and conquer algorithm for spectral sparsification, by tightly integrating the BC tree decomposition, aiming to reduce the runtime for computing the effective resistance values as well as to maintain the graph connectivity. We present two variations, called BC_SS (for spectral sparsification of *edges*) and BC_SV (for spectral sampling of *vertices*).

More specifically, we divide a big complex graph into a set of biconnected components, and then compute the spectral sparsification (i.e., effective resistance values) for each biconnected component in parallel. Namely, the *effective resistance* values of the edges are computed for each biconnected component, as a fast *approximation* of the effective resistance values of the original graph.

Let $G = (V, E)$ be a graph with a vertex set V ($n = |V|$) and an edge set E ($m = |E|$). The *adjacency matrix* of an n -vertex graph G is the $n \times n$ matrix A , indexed by V , such that $A_{uv} = 1$ if $(u, v) \in E$ and $A_{uv} = 0$ otherwise. The *degree matrix* D of G is the diagonal matrix where D_{uu} is the degree of vertex u . The *Laplacian* of G is $L = D - A$. The *spectrum* of G is the list $\lambda_1, \lambda_2, \dots, \lambda_n$ of eigenvalues of L . Suppose that we regard a graph G as an electrical network where each edge e is a $1-\Omega$ resistor, and a current is applied. The effective resistance $r(e)$ of an edge e is the voltage drop over the edge e , see [12].

3.1 Algorithm BC_SS

Let $G = (V, E)$ be a connected graph with a vertex set V and an edge set E , and let $G_i, i = 1, \dots, k$, denote biconnected components of G .

The BC_SS algorithm first computes the BC tree decomposition, and then adds the cut vertices and their incident edges to the spectral sparsification G' of G . This is due to the fact that the cut vertices play important roles in preserving the connectivity of the graph as well as in social network analysis, such as brokers or important actors connecting two different communities together.

Next, it computes a spectral sparsification G'_i for each biconnected component $G_i, i = 1, \dots, k$ of G . Specifically, for each component G_i , we compute the effective resistance values $r(e)$ of the edges, and then sample the edges with the largest effective resistance values. Finally, it merges $G'_i, i = 1, \dots, k$ to obtain the spectral sparsification G' of G . The BC_SS algorithm is described as follows:

Algorithm BC_SS

1. *Partitioning*: Divide a connected graph G into biconnected components, $G_i, i = 1, \dots, k$.
2. *Cut vertices*: Add the cut vertices and their incident edges to the spectral sparsification G' of G .
3. *Spectral sparsification*: For each component G_i , compute a spectral sparsification G'_i of G_i . Specifically, compute the effective resistance values $r(e)$ of the edges, and then sample the edges with largest effective resistance values.
4. *Aggregation*: Merge all G'_i of G_i to compute the spectral sparsification G' of the original graph G .

3.2 Algorithm BC_SV

The BC_SV algorithm is a divide and conquer algorithm that uses spectral sampling of vertices [5]: i.e., adapt the spectral sparsification approach, by sampling *vertices* rather than edges. More specifically, we define an *effective resistance* value $r(v)$ for each vertex v as the sum of effective resistance values of the incident edges, i.e., $r(v) = \sum_{e \in E_v} r(e)$, where E_v represents a set of edges incident to a vertex v .

The BC_SV algorithm first computes the BC tree decomposition, and then adds the cut vertices and their incident edges to the spectral sampling G' of G . Next, it computes a spectral vertex sampling G'_i for each biconnected component $G_i, i = 1, \dots, k$ of G . Specifically, for each component G_i , we compute the effective resistance values $r(v)$ of the vertices, and then sample the vertices with the largest effective resistance values. Finally, it merges $G'_i, i = 1, \dots, k$ to obtain the spectral sampling G' of G . The BC_SV algorithm is described as follows:

Algorithm BC_SV

1. *Partitioning*: Divide a connected graph G into biconnected components, $G_i, i = 1, \dots, k$.
2. *Cut vertices*: Add the cut vertices and their incident edges to the spectral sampling G' of G .
3. *Spectral vertex sampling*: For each component G_i , compute spectral vertex sampling G'_i of G_i . Specifically, compute the effective resistance values $r(v)$ of the vertices, and then sample the vertices with largest effective resistance values.

4. *Aggregation*: Merge all G'_i of G_i to compute the Spectral vertex sampling G' of the original graph G .

4 BC_SS and BC_SV Experiments

We design experiments to compare the Spectral Sparsification (SS) [2], BC_SS and Random Edge sampling (RE) (resp., Spectral Vertex sampling (SV) [5], BC_SV and Random Vertex sampling (RV), implemented in Java. Analysis of experimental results such as metrics and statistics are implemented in Python. All programs were run on a MacBook Pro with 2.2 GHz Intel Core i7, 16 GB 1600 MHz DDR3, and macOS Sierra version 10.12.6.

The main hypotheses of our experiments include:

- *H1*: BC_SS computes effective resistance values of edges faster than SS [2].
- *H2*: The effective resistance values and the rankings of edges (resp., vertices) computed by BC_SS (resp., BC_SV) are good approximations of those computed by SS (resp., SV), and their similarity increases with the sampling ratio.
- *H3*: Graph samples computed by BC_SS (resp., BC_SV) have almost the same sampling quality as SS (resp., SV), and significantly better than RE (resp., RV).
- *H4*: Graph samples computed by BC_SS (resp., BC_SV) produce almost the same visualization as SS (resp., SV).

The main rationale behind the hypotheses is that the graph samples computed by BC_SS and SS (resp., BC_SV and SV), are very similar, since the ranking of edges (resp., vertices) based on the resistance values are highly similar. We experiment with benchmark real world graphs [2] and synthetic data sets, see Table 1. The real world graphs are scale-free graphs with highly imbalanced size of biconnected components (i.e., big biconnected component). The synthetic graphs are generated with balanced size of biconnected components.

Table 1. Data sets

Graph	V	E
facebook	4039	88234
G4	2075	4769
G15	1789	20459
oflights	2939	14458
p2pG	8846	31839
soch	2426	16630
wiki	7115	100762
yeastppi	2361	6646

(a) Real world graphs

Graph	Abbr	V	E
<i>syn_path20_150_200_True</i>	sp20	3462	5410
<i>syn_tree4.5_10.10_100_True</i>	st4.5	21955	34957
<i>syn_tree4.9_10.10_30_True_2</i>	st4.9	18936	40411
<i>syn_tree6.3.4_10.30_True_2</i>	st6.3	11679	24868
<i>syn_tree6.3.4_10.30_True_3</i>	st6.3.3	13237	41936

(b) Synthetic graphs

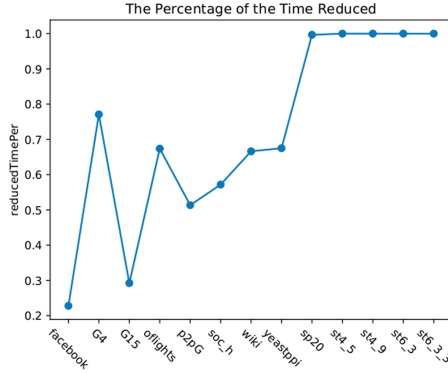


Fig. 1. Significant runtime improvement by BC_SS over SS.

4.1 Runtime Improvement

Figure 1 shows significant runtime improvement for computing effective resistance values by BC_SS over SS. The runtime improvement is much higher for the synthetic graphs, achieving above 99% on average, while the improvement on the real world graphs varies depending on their structures. For example, BC_SS improved 77% of the runtime for the G4 graph, while it improved 23% for the Facebook graph due to the existence of the giant biconnected component.

Overall, our experiments show that BC_SS is significantly faster than SS, confirming hypothesis H1.

4.2 Approximation on the Effective Resistance Values

Figures 2(a) and (b) show the mean of the differences in effective resistance values computed by SS and BC_SS with sampling ratio from 5% to 100%. Figures 2(c) and (d) show the mean of the differences in effective resistance values computed by SV and BC_SV with sampling ratio from 5% to 100%.

Overall, it clearly shows that the mean of the differences in effective resistance values computed by BC_SS and SS (resp., BC_SV and SV) is very small for most of the data sets, supporting hypothesis H2.

More specifically, for BC_SS, smaller than -0.0175 for real world graphs; smaller than -0.12 , with one outlier, for synthetic graphs. For BC_SV, smaller than -2.5 (compared to resistance values of edges, -2.5 is equivalent to -0.06) for real world graphs; smaller than -0.5 (equivalent to -0.08) for synthetic graphs.

Interestingly, in contrast to the runtime improvement results, real world graphs have a better similarity in the effective resistance values than synthetic graphs, due to the existence of the big giant component. Namely, the effective resistance values computed from the big biconnected component are very similar to the effective resistance values computed for the whole graph.

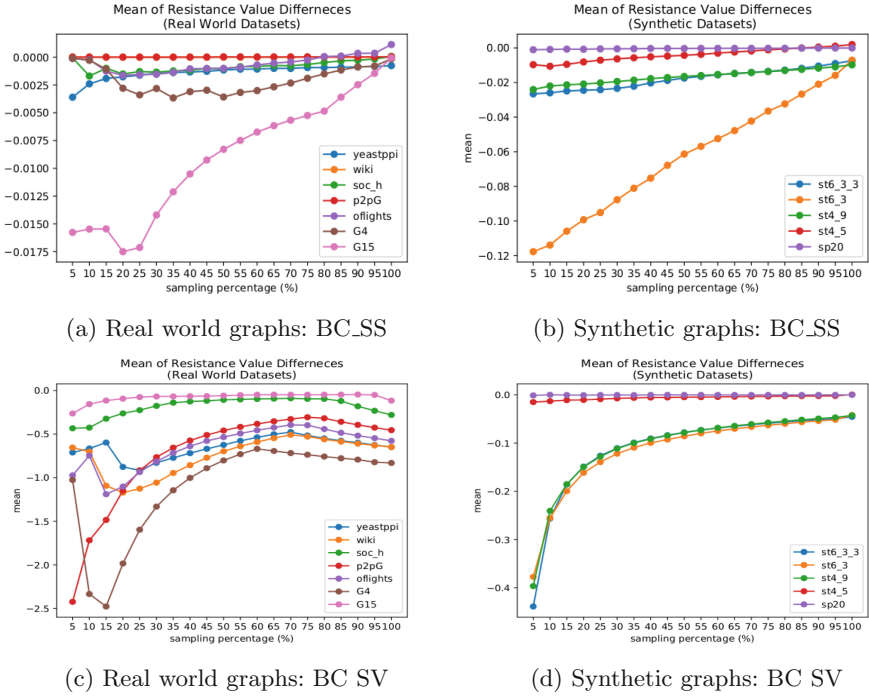


Fig. 2. The mean of the differences in effective resistance values computed by BC_SS and SS ((a), (b)); by BC_SV and SV ((c), (d)).

4.3 Approximation on the Ranking of Edges and Vertices

We define the *sampling accuracy* based on the proportion of the common sampled edges (resp., vertices) between two graph samples computed by SS and BC_SS (resp., SV and BC_SV). A high sampling accuracy indicates that both graph samples are highly similar. Namely, the sampling accuracy shows how well the effective resistance values computed by BC_SS (resp., BC_SV) can serve as a good approximation of the values computed by SS (resp., SV).

Figures 3(a) and (b) (resp., (c) and (d)) show the sampling accuracy of BC_SS (resp., BC_SV) with sampling ratio from 5% to 100%. It is easy to observe that for all data sets, *the sampling accuracy increases as the sampling ratio increases, supporting hypothesis H2*.

Specifically, for BC_SS, synthetic graphs perform better: they achieve above 50% sampling accuracy at sampling ratio 5%, and then quickly rise up to 80% at sampling ratio 15%, with steady improvement towards 100% as the sampling ratio increases.

The performance of BC_SS on the real world graphs shows different patterns, depending on their structure. Interestingly, the Facebook graph has excellent sampling accuracy at all sampling ratios, achieving above 80%, while it performs the worst on the runtime improvement and the difference in resistance

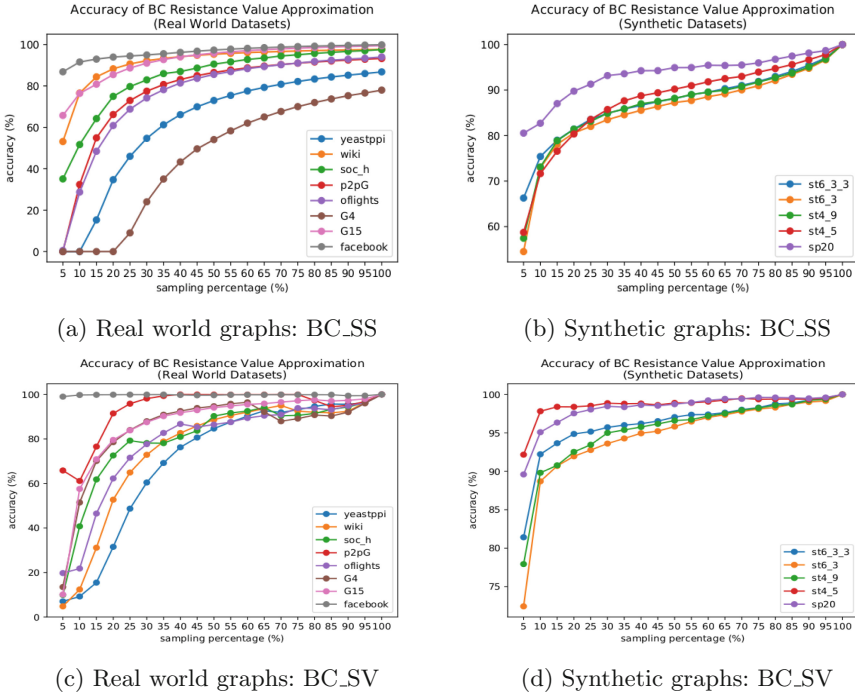


Fig. 3. The sampling accuracy of BC_SS ((a), (b)) and BC_SV ((c), (d)).

values. On the other hand, graph $G4$ shows very high performance on runtime improvement and the difference in resistance values is quite small, however the sampling accuracy is low when the sampling ratio is smaller than 20%.

For BC_SV, synthetic graphs show excellent performance overall, achieving above 70% sampling accuracy at sampling ratio 5%, and then rise up to 90% at sampling ratio 10%, with steady improvement towards 100% as the sampling ratio increases. For real world graphs, BC_SV achieves above 70% sampling accuracy at sampling ratio 20% for all graphs. Particularly, the Facebook graph shows excellent sampling accuracy at all sampling ratios, achieving above 99%.

Furthermore, the correlation analysis of the ranking of edges (resp., vertices) based on resistance values computed by SS and BC_SS (resp., SV and BC_SV) shows strong and positive results for all data sets. *Overall, our experiments and analysis confirm that BC_SS (resp., BC_SV) computes good approximations on the rankings of edges (resp., vertices) based on effective resistance values, compared to SS (resp., SV), validating hypothesis H2.*

4.4 Graph Sampling Quality Metrics Comparison

We use well known sampling quality metrics [6]: Degree Correlation (Degree), Closeness centrality (Closeness), Clustering coefficient (CC), and Average Neighbor Degree (AND). More specifically, we use the *Kolmogorov-Smirnov* (KS) distance value to compute the distance between two Cumulative Distribution

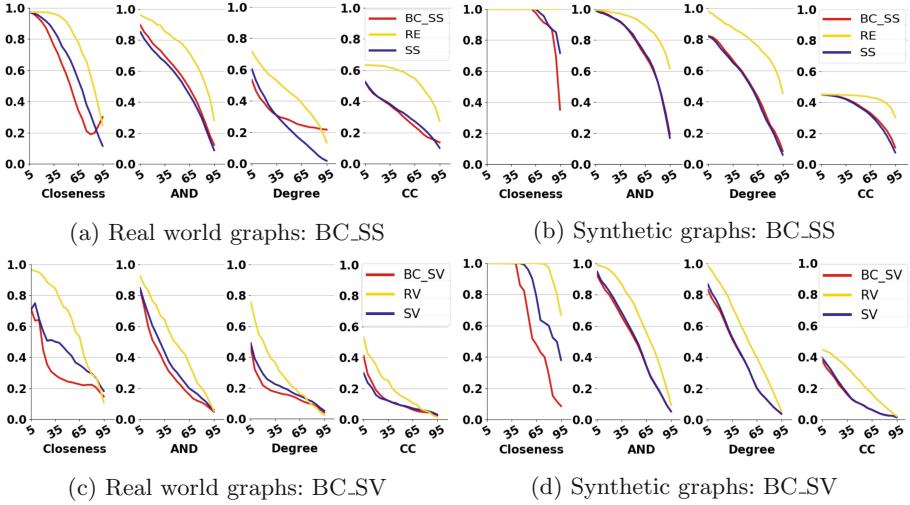


Fig. 4. The KS values of the sampling quality metrics (Closeness, AND, Degree, CC) of graph samples. (a), (b): computed by BC_SS, SS, RE; (c), (d): BC_SV, SV, RV. The lower KS value means the better result. BC_SS and SS (resp., BC_SV and SV) perform highly similar and better than RE (resp., RV). (Color figure online)

Functions (CDFs) [4]. The KS distance value is between 0 to 1: the lower KS value means the better result. Namely, the KS distance value closer to 0 indicates higher similarity between CDFs.

Figures 4(a) and (b) show the average (over all data sets) of the KS distance values of the graph samples computed by BC_SS (red), SS (blue), and Random Edge sampling (RE) (yellow), with four sampling quality metrics. Clearly, SS and BC_SS perform consistently better than RE for both types of graphs, as we expected. More importantly, the performance of SS and BC_SS are almost identical across all the metrics, especially for synthetic graphs. For the real world graphs, the performance of SS and BC_SS are highly similar on Closeness, AND, and CC metrics.

Figures 4(c) and (d) show the average (over all data sets) of the KS distance values of the graph samples computed by BC_SV (red), SV (blue), and Random Vertex sampling (RV) (yellow), with four sampling quality metrics. Clearly, SV and BC_SV perform consistently better than RV for both types of graphs, as we expected. More importantly, the performance of SV and BC_SV are almost similar on all the metrics, especially for AND, Degree, and CC. In particular, we observe that BC_SV shows the largest improvement on the Closeness metric, significantly better than SV.

In summary, our experimental results with sampling quality metrics confirm that both SS and BC_SS (resp., SV and BC_SV) outperform RE (resp., RV), and the graph samples computed by BC_SS (resp., BC_SV) have almost the same sampling quality as those computed by SS (resp., SV), confirming hypothesis H3.

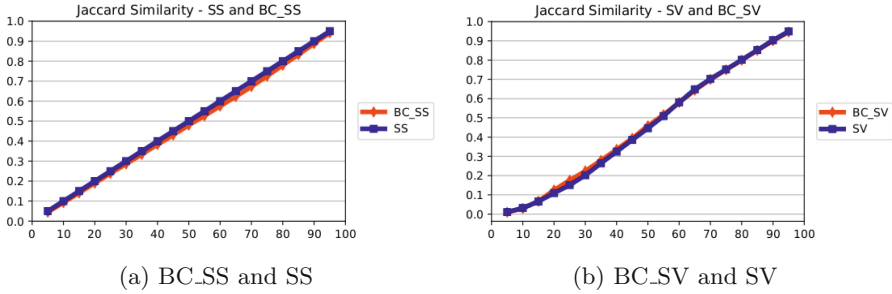


Fig. 5. Jaccard similarity index comparison of graph samples, computed by BC_SS and SS ((a), (b)) (resp., BC_SV and SV ((c), (d))): almost identical.

4.5 Jaccard Similarity Index Comparison

We also computed the Jaccard similarity index for testing similarity between the original graph G and the graph samples G' and G'' computed by SS and BC_SS (resp., SV and BC_SV). More specifically, it is defined as the size of the intersection divided by the size of the union of the two graphs (value 1 indicates that two graphs are the same).

Figure 5 shows the *average* Jaccard similarity values for real world graphs and synthetic graphs for BC_SS and SS (resp. BC_SV and SV), with sampling ratio from 5% to 95%. Clearly, for both data sets, *the Jaccard similarity index linearly increases with the sampling ratio, and SS and BC_SS (resp., SV and BC_SV) perform almost the same, validating hypothesis H3.*

4.6 Visual Comparison: SS vs. BC_SS and SV vs. BC_SV

We conduct visual comparison of graph samples computed by SS, BC_SS, SV, and BC_SV using the *Backbone* layout, specifically designed to untangle the hairball drawings of large graphs [10].

Figure 6 shows graph samples with sampling ratio at 20% for real world graphs (*facebook*, G_{15} , G_4 , *oflights*, *soc_h*, *yeastppi*) as well as a synthetic graph *st4_5*. *Visual comparison clearly shows that SS and BC_SS (resp., SV and BC_SV) produce almost identical visualizations, validating hypothesis H4.*

Overall, spectral edge sampling methods (SS and BC_SS) and spectral vertex sampling methods (SV and BC_SV) produce visually highly similar graph samples. For some cases, the density of graph samples are slightly different, depending on the density of the original graphs. For example, for graphs G_4 and *yeastppi*, spectral vertex sampling methods (SV and BC_SV) compute graph samples which better captures the dense structure of the original graph.



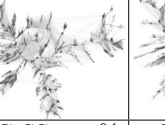
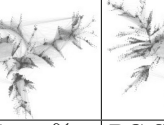




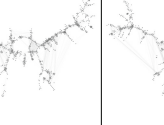
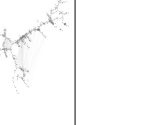
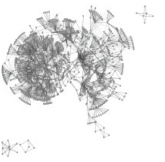
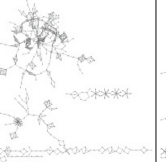
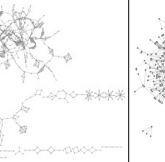
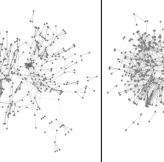
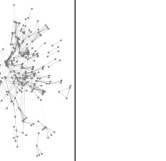
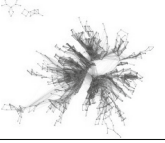
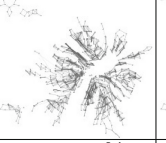
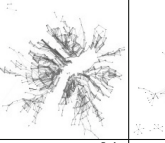
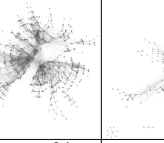
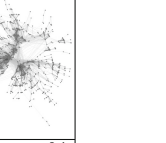
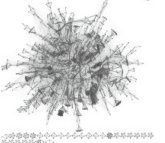
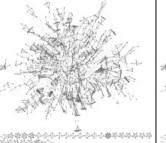
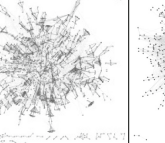
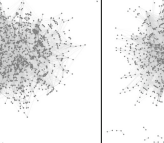
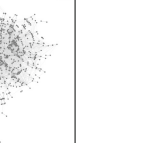
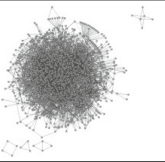
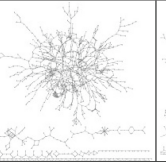
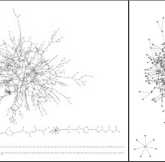
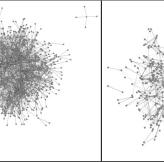
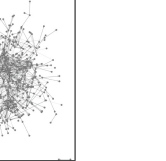
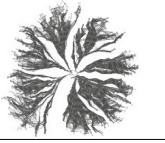
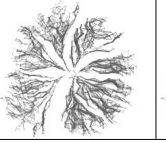
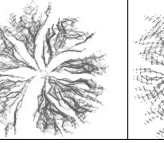
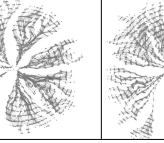
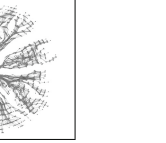
Original - <i>facebook</i>	SS - 20%	BC_SS - 20%	SV - 20%	BC_SV - 20%
				
Original - <i>G15</i>	SS - 20%	BC_SS - 20%	SV - 20%	BC_SV - 20%
				
Original - <i>G4</i>	SS - 20%	BC_SS - 20%	SV - 20%	BC_SV - 20%
				
Original - <i>oflights</i>	SS - 20%	BC_SS - 20%	SV - 20%	BC_SV - 20%
				
Original - <i>soc_h</i>	SS - 20%	BC_SS - 20%	SV - 20%	BC_SV - 20%
				
Original - <i>yeastppi</i>	SS - 20%	BC_SS - 20%	SV - 20%	BC_SV - 20%
				
Original - <i>st4_5</i>	SS - 20%	BC_SS - 20%	SV - 20%	BC_SV - 20%
				

Fig. 6. Comparison of graph samples of real world graphs and a synthetic graph with 20% sampling ratio, computed by SS, BC_SS, SV and BC_SV. SS and BC_SS (resp. SV and BC_SV) produce almost identical visualizations.

5 Conclusion and Future Work

In this paper, we present two new spectral sampling methods, BC_SS and BC_SV, tightly integrating the BC tree decomposition for fast computation and spectral sparsification to obtain high quality graph samples, preserving structural properties of graphs. Extensive experimental results with both real world graphs and synthetic graphs demonstrate that our new BC tree-based spectral sampling approach is significantly faster than existing methods, while preserving highly similar quality sampling results, based on the comparison of resistance values, rankings of edges/vertices, graph sampling quality metrics, Jaccard similarity index, and visual comparison.

For future work, we plan to design new graph sampling methods for big graph visualization, by combining other graph partitioning methods. For example, see [8] for an edge sampling method integrating spectral sparsification with the decomposition of biconnected graphs into triconnected components.

References

1. Barrat, A., Barthélemy, M., Pastor-Satorras, R., Vespignani, A.: The architecture of complex weighted networks. *Proc. Natl. Acad. Sci. U.S.A.* **101**(11), 3747–3752 (2004)
2. Eades, P., Nguyen, Q.H., Hong, S.: Drawing big graphs using spectral sparsification. In: *Proceedings of GD 2017*, pp. 272–286 (2017)
3. Freeman, L.C.: Centrality in social networks conceptual clarification. *Soc. Netw.* **1**(3), 215–239 (1978)
4. Gammon, J., Chakravarti, I.M., Laha, R.G., Roy, J.: *Handbook of Methods of Applied Statistics* (1967)
5. Hu, J., Hong, S., Eades, P.: Spectral vertex sampling for big complex graphs. In: *Proceedings of Complex Networks*, pp. 216–227. Springer (2019)
6. Hu, P., Lau, W.C.: A survey and taxonomy of graph sampling. *CoRR* abs/1308.5865 (2013)
7. Leskovec, J., Faloutsos, C.: Sampling from large graphs. In: *Proceedings of SIGKDD*, pp. 631–636. ACM (2006)
8. Meidiana, A., Hong, S., Huang, J., Eades, P., Ma, K.: Topology-based spectral sparsification. In: *Proceedings of LDAH*, pp. 73–82. IEEE (2019)
9. Newman, M.E.J.: Mixing patterns in networks. *Phys. Rev. E* **67**(2), 26126 (2003)
10. Nocaj, A., Ortman, M., Brandes, U.: Untangling the hairballs of multi-centered, small-world online social media networks. *JGAA* **19**(2), 595–618 (2015)
11. Saramäki, J., Kivelä, M., Onnela, J.P., Kaski, K., Kertész, J.: Generalizations of the clustering coefficient to weighted complex networks. *Phys. Rev. E* **75**(2), 27105 (2007)
12. Spielman, D.A., Teng, S.H.: Spectral sparsification of graphs. *SIAM J. Comput.* **40**(4), 981–1025 (2011)
13. Wu, Y., Cao, N., Archambault, D.W., Shen, Q., Qu, H., Cui, W.: Evaluation of graph sampling: a visualization perspective. *IEEE TVCG* **23**(1), 401–410 (2017)



Graph Signal Processing on Complex Networks for Structural Health Monitoring

Stefan Bloemheuv^{1,2}(✉), Jurgen van den Hoogen^{1,2}, and Martin Atzmueller³

¹ Tilburg University, Tilburg, The Netherlands

² Jheronimus Academy of Data Science, 's-Hertogenbosch, The Netherlands
{s.d.bloemheuv¹, j.o.d.hoogen}@jads.nl

³ Semantic Information Systems Group, Osnabrück University, Osnabrück, Germany
martin.atzmueller@uni-osnabrueck.de

Abstract. In this work, we demonstrate the application of a framework targeting *Complex Networks* and *Graph Signal Processing* (GSP) for *Structural Health Monitoring* (SHM). By modeling and analyzing a large bridge equipped with strain and vibration sensors, we show that GSP is capable of selecting the most important sensors, investigating different optimization techniques for selection. Furthermore, GSP enables the detection of graph signal patterns (mode shapes), grasping the physical function of the sensors in the network. Our results indicate the efficacy of GSP on complex sensor data modeled in complex networks.

Keywords: Complex networks · Graph signal processing · Sensor data · Networks for physical infrastructures · Structural health monitoring

1 Introduction

For several domains and problems, complex networks provide natural means for representing complex data, e. g., concerning multi-relational data, dynamic behavior, and complex systems in general. In particular, targeting dynamic, temporal, and continuous sequential data, *Graph Signal Processing* (GSP) [24] has emerged as a prominent and versatile framework for analyzing such complex data. This concerns both the analysis of network structure as well as its dynamics. GSP extends on classical signal processing by including irregular structures such as Graphs/Networks [22]. The advantage of graphs over classical data representations is that graphs naturally account for such irregular relations [25].

In this paper, a computational framework utilizing GSP for the analysis of complex sensor data represented in complex networks is presented. Our application context is given by Structural Health Monitoring (SHM), a data-driven diagnostic framework for investigating and estimating the integrity of massive structures [1,23]. It aims at improving safety, reliability, efficiency, and

(cost-)effectiveness in civil infrastructures such as pipeline systems, buildings, and bridges. To the best of the authors' knowledge, this is the first time that GSP has been applied for such a data modeling and analysis task of networks for real-world physical infrastructures.

SHM is a multidisciplinary field that combines insights from civil engineering, signal processing, sensor technology, and data mining [16]. Most often, data that feeds SHM systems comes in the form of discrete-domain signals (time series). We apply GSP for SHM in the context of an application from a Dutch SHM project, called *InfraWatch* [12]. The real-world data used is captured by a sensor system installed on a major highway bridge called the *Hollandse Brug*. Then, sensors monitor pressure of traffic passing over the bridge. Essentially, the basic principle for SHM is that global parameters (mode shapes, natural frequencies) are functions of physical properties such as mass, damping, and stiffness [7, 21]. Mode shapes can be considered as specific patterns where signals and their frequencies are put into different modal categories (from a signal processing view). Vibration-based sensors can detect characteristic parameters such as frequency, mode shape curvatures, and flexibility. Strain-based sensors rely on the assumption that changes in the physical properties will reflect in the amplitudes of strain measures. Both local and global characteristics can then be extracted. For example, local deviations of sensors can indicate faulty sensor readings. Global characteristics could assess the overall change in stiffness of a structure [21], or calculate the maximum structural capacity of a bridge [21].

Another specific problem concerns resource-aware techniques for SHM, i. e., identifying the minimal subset of sensors capable of reconstructing the signal using GSP. This can then be applied for optimizing sensor networks [6], e. g., minimizing the needed number of sensors for monitoring the bridge. Furthermore, GSP enables the identification of certain events as well as the detection of specific patterns in complex data. For SHM in our bridge scenario, this concerns, for example, the detection of traffic peaks, as well as specific patterns observed when a large amount of pressure is exerted on distinct parts of the bridge. These patterns lead to direct hints indicating the health of the bridge [21]. Both of these problems are also investigated in this paper, i. e., how modeling and analysis are to be performed and to what extent we can identify such subsets of sensors and patterns, respectively.

Our contributions are summarized as follows:

1. We propose a computational framework applying GSP for SHM, covering network modeling, GSP, and subsequent analysis.
2. We demonstrate the application of this framework in a case study utilizing a real-world dataset of rich sensor data modeled in a complex network.
 - (a) We propose the modeling options taken for making the real-world dataset applicable for GSP using a complex network representation.
 - (b) We present comprehensive analysis results, regarding sensor network modeling in a resource-aware way, aiming towards a minimal set of sensors for reconstructing the given signals.
 - (c) We provide modeling results on signal pattern and event identification.

The rest of the paper is structured as follows: Sect. 2 provides an overview of GSP and introduces necessary theoretical notions. Next, Sect. 3 presents our proposed framework and describes the methodology in detail. After that, Sect. 4 presents the case study and discusses our results. Finally, Sect. 5 concludes with a summary and provides interesting directions for future work.

2 Background on GSP

This section first introduces basic concepts of signal processing on graphs and the necessary theoretical background. For a detailed overview on GSP see e. g., [18, 24].

Overview. Traditional signal processing can be extremely powerful in uniform, euclidean domains such as sampled audio or power circuits. However, not all domains have such a desirable property. For example, when the data at hand are sensors placed along specific locations in a building, the topography will most likely not resemble a uniform square grid. Specifically, there could be walls that influence the positions of sensors at each floor (and strength of the signal), or there could be floors without any sensors at all. Thus, the complexity of such networks implies that the data coming from irregular and complex structures do not lend themselves for standard tools [18]. This motivates more complex modeling, e. g., by including spatial dimensions, leading towards GSP; it extends Signal Processing by including irregular structures modeled as Graphs [22]. Intuitively, signal data on a graph can then be visualized as a finite set of samples, with samples assigned to each node of the graph.

GSP: Basic Definitions Formally, a graph is defined as $G = (V, E)$ where V are the nodes and E the edges. The graph G can be represented with the laplacian matrix $L \in \mathbb{R}^{N \times N}$ where $|V| = N$, which is the degree matrix minus the adjacency matrix [24].

- A graph signal is defined by associating real data values s_n to each vertex. In vector notation, a graph signal can be written as $s = [s_0, s_1, \dots, s_{N-1}]^T \in \mathbf{R}$.
- In Digital Signal Processing, a signal shift is implemented as a shift in time of length N , resulting in $\hat{s} = s_{n-1}$. In GSP, a shift is defined as a local operation that replaces a signal value by a combination of the values connected to V_n weighted by their respective edge weights. The two most popular graph shift operators are the laplacian and adjacency matrix.
- An important transformation in classical Signal Processing is the Fourier transform. In terms of GSP, the Graph Fourier Transform (GFT) converts the graph signal from the vertex domain into the graph spectral domain. GSP achieves this transformation by spectral decomposition of

$$L = V\Lambda V^{-1}, \quad (1)$$

where the columns v_n of the matrix V are the eigenvectors of the laplacian L , and Λ the diagonal matrix of the corresponding eigenvalues. The eigenvalues

act as the frequencies on the graph [20]. The GFT of a signal s is then computed as $\hat{s} = U^*s$ where U is the Fourier basis of a graph and U^* the conjugate transpose of U . The inverse GFT of a Fourier domain signal \hat{s} is defined as $s = U\hat{s}$ where U is the Fourier basis of a graph.

- Similar to classical signal processing, filters can be applied to graph signals. The fundamental idea is to transform the graph signal into the graph spectral domain, weaken unwanted frequencies or magnify wanted frequencies of the signal by altering the Fourier coefficients, and convert the signal back to the vertex domain.

3 Method

In this section, an overview of our analysis framework is provided. After that, the dataset and network modeling techniques are described.

3.1 Overview: GSP Methodological Framework

Figure 1 shows an overview of the proposed framework applying GSP for SHM using complex networks. Overall, the framework provides an incremental and iterative methodology for analysis. The complex signal (sensor network data) is first modeled into a *complex network*, which can be assessed in a semi-automatic approach to apply refinements and enrichment of the network (or the respective data, e. g., when defective sensors are detected). After that, GSP is applied on the network to obtain a specific *GSP model* which can be utilized and deployed for SHM, e. g., for identifying a minimal subset of sensors, or for detecting specific patterns, events, mode shapes, etc.

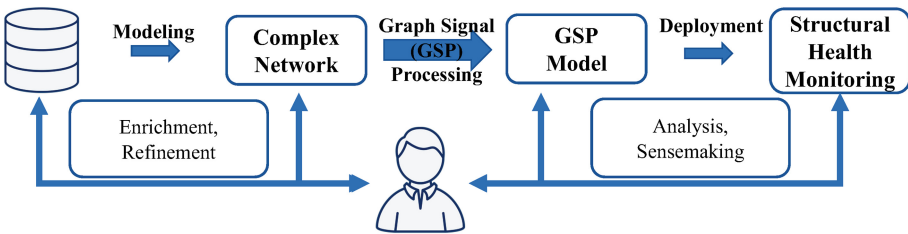


Fig. 1. Overview on the proposed Graph Signal Processing framework.

3.2 Dataset

The *InfraWatch* project investigated an important highway bridge in the Netherlands: the *Hollandse Brug* (built in 1969) – connecting the provinces Flevoland and Noord-Holland. Sensors were placed on the bridge in 2007 since reports

indicated that the bridge did not meet the quality and security requirements. The network consists of 145 sensors placed along the width of the bridge, which include 34 vibration, 50 horizontal strain (X-strain), 41 vertical strain (Y-strain), and 20 temperature sensors. The dataset has been used for various data mining techniques, such as time series analysis [26] and modal analysis [17].

Overall, the dataset made available to us consists of 5 min of high-resolution sensor data, about 30,000 observations in total. It contains several traffic events, of which the 10 most significant events are investigated. For data preprocessing and alignment, the domain expert mentioned that the strain sensors were not scaled on the same range and that clock times of the sensors were not aligned; time synchronization is a general open challenge [15], regarding simultaneous data collection on all sensors. Thus, the clock times of the sensors were aligned by matching the peaks in the sensor readings, and the data were rescaled by z-score standardization. Lastly, the average values per 100ms were taken from the original data (sampled at 100 Hz) to smooth out the signal. The sensors were placed at three cross-sections within one span (see [16]). Thus, to make the connections in the network meaningful, the 31 sensors in the middle and right cross-sections were dropped. Also, 4 sensors were found defective, which reduced the total set of sensors from 145 to 110 (Fig. 2).



Fig. 2. Sensor locations at one of the girders of the bridge (from [16]). Sensors are either embedded or attached to the deck or girders.

3.3 Network Creation

To create a sensor network from the bridge data, the x and y locations of each sensor were extracted from the bridge blueprint. For GSP modeling the most crucial step is determining what each edge (i, j) should resemble [14]. An option would be using geographical distance, but that would not grasp the (functional) relationships between the sensors. Because the bridge contains girders that capture most of the strain, sensors at the top of the bridge should, while indeed being placed geographically close to the sensors at the girders, act in the exact opposite

of the strain sensors at the girders. Therefore, the edges were determined by the correlation score between the sensor measurements, selecting the top-3 edges for modeling (excluding vibration sensors, which kept all their original edges).

To conclude, four networks are created based on the given set of desired sensors: X-strain ($|V| = 42$, $|E| = 126$), Y-strain ($|V| = 37$, $|E| = 111$), X-Y combined ($|V| = 79$, $|E| = 237$) and Vibration ($|V| = 15$, $|E| = 26$). The strain sensors are used in the analysis of sampling and mode shape identification, while the vibration sensors only assisted in identifying mode shapes.

3.4 Node Subset Selection – Sensor Subset Sampling

A fundamental task in GSP is to infer the values of certain sensors by interpolating them from a sample, e. g., when the application requires cost or bandwidth constraints limiting the number of nodes that can be observed.

In this paper, sampling is applied by finding the optimal subset of sensors that can reconstruct the original signal at a certain time point. The time points are *specified* as the moments in the signal where events take place, since it does not make sense to incorporate the error rates at time points where no traffic event occurs. Since a brute-force approach is not feasible, the following strategies are investigated: random search and hill-climbing. Both techniques are very common in the greedy search optimization literature, which tries to approximate a solution for the known NP-complete combinatorial problem for large values of p and N [2, 4, 13, 19]. The random search acts as a baseline and generates a random set of sensors that are sampled. For the hill-climbing technique, we propose two specific strategies: top-down and bottom-up. In the top-down strategy (Forward Selection), the algorithm starts with all the sensors and eliminates one-by-one the least informing sensors. In the bottom-up strategy (Backward Elimination), the algorithm starts with zero sensors selected and gradually selects sensors based on the most decrease in error. Both greedy techniques contain a random element by picking from the top-3 best or worst performing sensors (depending on the algorithm) in each iteration, which helps to combat local maxima and minima. Each algorithm ran for 500 iterations and an iteration stopped once the desired number of sensors were selected (25% of the $|V|$ sensors respectively), and only unique solutions were considered successful.

For evaluating the subset of sensors and estimating the (total) signal from this subset of sensors, we apply Tikhonov Minimization (see [8, 22]) in each iteration of the sampling procedure to reconstruct the entire signal. The function solves for the unknown vector x :

$$\arg \min_x \|Mx - y\|_2^2 + \tau x^T Lx, \quad (2)$$

if $\tau > 0$ and

$$\arg \min_x x^T Lx : y = Mx, \quad (3)$$

otherwise, where y is the graph signal, M is the masking vector which resembles the nodes that are sampled, L the laplacian matrix, and τ the regularization

parameter. Considering the Tikhonov Minimization, several values of the regularization parameter τ were applied. However, the default value of $\tau = 0$ was used since this yielded the best results.

Lastly, each algorithm was tested on low-pass filtered data $g(x) = \frac{1}{1+0.5 \cdot x}$. A graph filter is defined as a function over the graph frequencies, altering the graph frequency content as a point-wise multiplication in the graph Fourier domain [11]. After the signals have been filtered, the Inverse Graph Fourier Transformation of the Fourier domain signal returns the signal in the time domain.

4 Results and Discussion

Below, we first present and discuss the results of sampling the sensors for obtaining a minimal subset of sensors which allows to reconstruct the total signal. After that, applications for mode shape identification will be discussed.

4.1 Sampling: Selecting a Minimal Subset of Sensors

Table 1 shows the Root Mean Squared Error (RMSE) for different conditions during the 10 most noticeable traffic events in the time series. We chose RMSE as a standard metric since it measures in the same unit as the variable of interest.

The domain expert indicated that Y-strain is harder to model since the bridge can move more freely in the Y-direction than the X-direction. Therefore, the algorithms perform best on the X-strain sensors but struggle with the Y-strain sensors. When directly comparing the algorithms, the top-down algorithm consistently outperforms the random (+29.92%) and bottom-up (+11.85%) algorithms in terms of RMSE. Also, the random algorithm was tested in a separate experiment for 50.000 iterations (100× more than the initial setting). When we consider the individual events, even after that many runs, the random algorithm consistently did not find any better solution than both hill-climbers. In that sense, running only *one* top-down iteration already outperforms very many random iterations (for any reasonable N of iterations).

Table 1. Mean and standard deviation of the best RMSE scores for each algorithm during all significant traffic events in the dataset. Non-filtered and Filtered stands for the conditions if a graph filter was applied to the signal or not.

Sensor type		Non-filtered			Filtered		
		X-strain	Y-strain	Combined	X-strain	Y-strain	Combined
Algorithm	Random	0.80	1.36	1.12	0.45	0.86	0.68
		(.32)	(.95)	(.76)	(.29)	(.62)	(.46)
	Top-down	0.60	1.06	0.74	0.31	0.66	0.38
		(.24)	(.75)	(.52)	(.19)	(.51)	(.29)
	Bottom-up	0.68	1.08	0.88	0.34	0.71	0.46
		(.30)	(.80)	(.63)	(.21)	(.53)	(.35)

Figures 3 and 4 show the performance of the X-strain sensors during the heaviest traffic event data (event 1). In comparison, the random algorithm ($M = 3.90$, $SD = 1.09$) performs poor in general. The bottom-up algorithm ($M = 2.60$, $SD = .45$) already improves from the random algorithm by a high margin; the differences in both distributions showed to be significant (as estimated via a t-test of the result from 500 iterations for each algorithm ($p < .001$)). The top-down algorithm ($M = 1.20$, $SD = .06$) shares no overlap with the bottom-up algorithm.

The top-down algorithm also shows a much smaller standard deviation, which indicates that it performs more consistent. Such behavior can be explained by the underlying procedures of the hill-climbers. The bottom-up algorithm finishes more iterations because it decides which nodes are selected instead of dropped. It makes decisions about the 25% selected sensors, while the top-down algorithm decides on the 75% sensors *not* selected. If a weak performing sensor is not dropped in the first few iterations, it will most likely be dropped in a later iteration since it will stay in the pool of sensors to drop *longer*. Therefore, a good strategy appears to be running a few top-down trials.

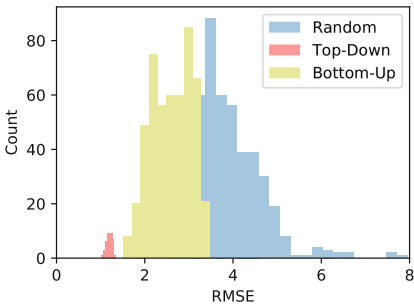


Fig. 3. RMSE scores of each algorithm on X-strain sensors during traffic event 1.

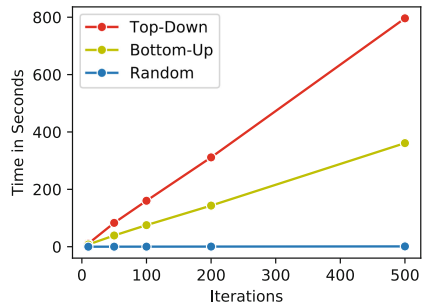


Fig. 4. Runtimes of each algorithm on X-strain sensors.

When examining the selected sensors by the top-down algorithm in Fig. 5, a nearly symmetrical selection performs most optimal. Such a pattern is especially visible in the X-strain sensors. These results indicate a hint of over-engineering in the number of sensors placed on the bridge. In addition, it is surprising that the second-lowest row (the sensors placed at a height of 10 in Fig. 5) of X-strain sensors was not sampled at all. This indicates that the sensors placed in the middle of the girders are obsolete when applying GSP.

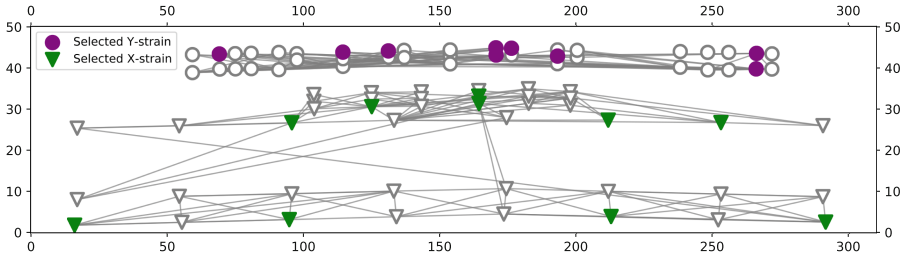


Fig. 5. The green and purple nodes are the selected sensors from the top-down algorithm. Circle-shaped nodes indicate the Y-strain sensors and triangle-shaped nodes indicate the X-strain sensors (X-strain/Y-strain are vertically separated).

4.2 Network Representation Example: Girders and Deck

Figures 6 and 7 show the X-strain sensors and Y-strain sensors placed on top of each other. Such behavior is expected since the bottom part of the bridge contains girders that carry most of the weight. Figure 7 shows traffic event 1 where strain is visible on the bottom right side of the bridge, indicating that some vehicle crossed by. The figure also shows a decrease in strain visible at the top of the bridge, of which our domain expert suggested that the girders perform their work correctly. Engineers could monitor the signals at each time point and assess how the strain and vibrations are distributed across the bridge.

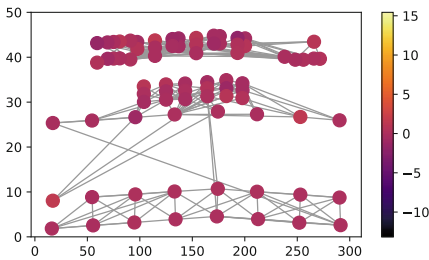


Fig. 6. The sensor network consisting of X (bottom component) and Y (top component) sensors. X-strain/Y-strain sensors are vertically separated for explanatory purposes).

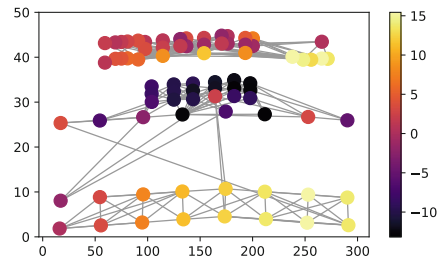


Fig. 7. The sensor network during traffic event 1. We can observe increased strain in the girders at the bottom right part of the bridge and the upper region of the Y-strain sensors, respectively.

4.3 Identification of Mode Shapes

To identify mode shapes, normally the frequencies of signals are categorised into a combination of different modes with Finite Element Method (FEM), a numerical technique for solving partial differential equations. FEM can be applied on

any physical phenomenon, e. g., wave propagation, fluid behavior and heat flow. FEM solves a problem by reducing a system in smaller parts called finite elements, which in turn form a mesh of the object. Each element contains a simple equation that when assembled, models the entire problem.

With GSP, this procedure is not always necessary since certain mode shapes can be spotted by examining the graph for a period of t time points. Figure 8a shows a combination of mode shapes that can be spotted in Fig. 8c. The bridge is vibrating back and forth, of which a time point where the left side of the bridge if decreasing in terms of vibration is shown. A supplementary page¹ with animated GIFs is available since static images do not show the full story.

Figure 8d shows a vehicle passing by on the right side of the bridge, and how the girders on the bottom right of the bridge carry the weight and allow the other parts of the bridge to decrease in strain level (see the left side of Fig. 8b). In future work, methods to automatically label moments in time by their combination of mode shapes could be investigated.

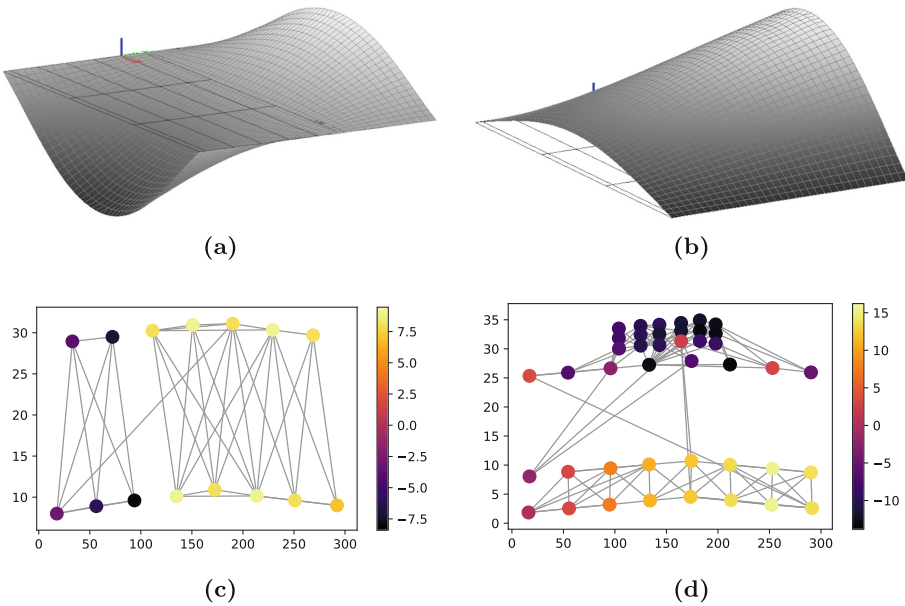


Fig. 8. (a) shows a FEM-based (from [17]) combination of mode shapes and the corresponding graph signal in (c). (b) shows a FEM-based (from [17]) combination of torsional mode shapes in the girders and the corresponding graph signal in (d).

¹ [Link to github page.](#)

5 Conclusions

In this work, we presented the application of a computational framework utilizing Graph Signal Processing for the analysis of complex sensor data in the domain of Structural Health Monitoring. In the proposed framework, GSP showed to be a promising technique to work with real-world complex sensor data. The results indicate that GSP is capable of selecting the most important sensors in the *Hollandse Brug*, a large bridge in the Netherlands, to arrive at a minimal subset of sensors in a resource-aware way. The top-down algorithm performed best of the tried algorithms. Using GSP for sensor selection could lead to significant cost-reductions in monitoring large civil infrastructures. Furthermore, the sensor selection could be used to increase the lifetime of battery-powered sensor networks, e.g., by calculating two optimal sets of sensors to turn on and off interchangeably.

In addition, our case study demonstrated a technique to find a combination of mode shapes in the graph signal plots, indicating important events/mode shapes in our application context. The events and mode shapes visible in the network can be used to assess the structural health of the bridge, since the mode shapes hint to other aspects of the bridge, such as stiffness and damping. Moreover, our GSP approach requires less modeling assumptions and engineering knowledge (e.g., building a complex FEM model).

For future research, we intend to investigate ways to detect mode shapes with GSP in an unsupervised manner, e.g., adapting/refining methods from anomaly detection [3,5]. Furthermore, we plan to apply GSP to other civil infrastructure datasets with applications domains such as, e.g., heat diffusion or fluid flow. Additionally, we aim to assess more data-driven methods in order to bypass the usage of greedy strategies, such as graph neural network approaches, e.g., [9, 27], also including combinations of other network analysis and GSP methods, e.g., utilizing more information as modeled in feature-rich networks [10]. Such methods could then as well be tested on other civil infrastructure datasets.

Acknowledgement. We wish to thank Dr. A.J. Knobbe for his domain knowledge considering the InfraWatch project, which he managed for several years.

References

1. Abdulkarem, M., Samsudin, K., Rokhani, F.Z., A Rasid, M.F.: Wireless sensor network for structural health monitoring: a contemporary review of technologies, challenges, and future direction. *Struct. Health Monit.* **19**(3), 693–735 (2020)
2. Aggarwal, C.C., Bar-Noy, A., Shamoun, S.: On sensor selection in linked information networks. *Comput. Netw.* **126**, 100–113 (2017)
3. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description. *Data Min. Knowl. Disc.* **29**(3), 626–688 (2015)
4. Anis, A., Gadde, A., Ortega, A.: Efficient sampling set selection for bandlimited graph signals using graph spectral proxies. *IEEE Trans. Signal Process.* **64**(14), 3775–3789 (2016)

5. Atzmueller, M., Arnu, D., Schmidt, A.: Anomaly detection and structural analysis in industrial production environments. In: Proceedings of the International Data Science Conference (IDSC 2017), Salzburg, Austria (2017)
6. Capellari, G., Chatzi, E., Mariani, S.: Cost-benefit optimization of structural health monitoring sensor networks. *Sensors* **18**(7), 2174 (2018)
7. Cornwell, P., Farrar, C.R., Doebling, S.W., Sohn, H.: Environmental variability of modal properties. *Exp. Tech.* **23**(6), 45–48 (1999)
8. Defferrard, M., Martin, L., Pena, R., Perraudin, N.: PyGSP: graph signal processing in python. <https://github.com/epfl-lts2/pygsp/>
9. Han, Z., Wang, Y., Ma, Y., Günnemann, S., Tresp, V.: Graph hawkes network for reasoning on temporal knowledge graphs. *CoRR* abs/2003.13432 (2020)
10. Interdonato, R., Atzmueller, M., Gaito, S., Kanawati, R., Langeron, C., Sala, A.: Feature-rich networks: going beyond complex network topologies. *Appl. Netw. Sci.* **4**(4), 1–13 (2019)
11. Isufi, E.: Graph-time signal processing: filtering and sampling strategies. Ph.D. thesis, Doctoral Thesis. Technische Universiteit Delft (2019)
12. Knobbe, A., Blockeel, H., Koopman, A., Calders, T., Obladen, B., Bosma, C., Galenkamp, H., Koenders, E., Kok, J.: Infracatch: data management of large systems for monitoring infrastructural performance. In: Proceedings of the International Symposium on Intelligent Data Analysis, pp. 91–102. Springer (2010)
13. Krause, A., Singh, A., Guestrin, C.: Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.* **9**(Feb), 235–284 (2008)
14. Mateos, G., Segarra, S., Marques, A.G., Ribeiro, A.: Connecting the dots: identifying network structure via graph signal processing. *IEEE Signal Process. Mag.* **36**(3), 16–43 (2019)
15. Mechitov, K., Kim, W., Agha, G., Nagayama, T.: High-frequency distributed sensing for structure monitoring. In: Proceedings of the First International Workshop on Networked Sensing Systems (INSS 2004), pp. 101–105 (2004)
16. Miao, S.: Structural health monitoring meets data mining. Ph.D. thesis, Doctoral Thesis. Leiden Institute of Advances Computer Science (2014)
17. Miao, S., Veerman, R., Koenders, E., Knobbe, A.: Modal analysis of a concrete highway bridge: structural calculations and vibration-based results. In: Proceedings of the Conference on Structural Health Monitoring of Intelligent Infrastructure, Hongkong (2013)
18. Ortega, A., Frossard, P., Kovačević, J., Moura, J.M., Vandergheynst, P.: Graph signal processing: overview, challenges, and applications. *Proc. IEEE* **106**(5), 808–828 (2018)
19. Puy, G., Tremblay, N., Gribonval, R., Vandergheynst, P.: Random sampling of bandlimited signals on graphs. *Appl. Comput. Harmon. Anal.* **44**(2), 446–475 (2018)
20. Sandryhaila, A., Moura, J.M.: Discrete signal processing on graphs: frequency analysis. *IEEE Trans. Signal Process.* **62**(12), 3042–3054 (2014)
21. Seo, J., Hu, J.W., Lee, J.: Summary review of structural health monitoring applications for highway bridges. *J. Perform. Constr. Facilit.* **30**(4), 04015072 (2016)
22. Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P.: The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.* **30**(3), 83–98 (2013)
23. Sony, S., Laventure, S., Sadhu, A.: A literature review of next-generation smart sensing technology in structural health monitoring. *Struct. Control Health Monit.* **26**(3), e2321 (2019)

24. Stankovic, L., Mandic, D., Dakovic, M., Brajovic, M., Scalzo, B., Constantinides, T.: Graph signal processing—part I: graphs, graph spectra, and spectral clustering. arXiv preprint [arXiv:1907.03467](https://arxiv.org/abs/1907.03467) (2019)
25. Stankovic, L., Mandic, D.P., Dakovic, M., Kisil, I., Sejdic, E., Constantinides, A.G.: Understanding the basis of graph signal processing via an intuitive example-driven approach [lecture notes]. *IEEE Signal Process. Mag.* **36**(6), 133–145 (2019)
26. Vespier, U., Knobbe, A., Nijssen, S., Vanschoren, J.: MDL-based analysis of time series at multiple time-scales. In: *ECML PKDD*, pp. 371–386. Springer (2012)
27. Zhang, Z., Cui, P., Zhu, W.: Deep learning on graphs: a survey. *IEEE Trans. Knowl. Data Eng.* (2020)



An Analysis of Four Academic Department Collaboration Networks with Respect to Gender

Lauren Nakamichi, Theresa Migler^(✉), and Zoë Wood

California Polytechnic State University, San Luis Obispo, CA, USA
tmigler@calpoly.edu

Abstract. Understanding how research is conducted is important for understanding how we get the information that influences the decisions we make. One method used for analyzing the patterns behind research is building collaboration networks. In a collaboration network, vertices represent researchers and an edge exists between two vertices if the corresponding researchers have collaborated.

We construct a collaboration network for the faculty in the Biology, Computer Science, Electrical Engineering, and Mathematics departments at a large undergraduate university in California. For each of these departments, we analyze collaboration patterns with respect to gender within this network. We find interesting collaborative behavior in the Mathematics department that differs from previously established claims about gender with respect to collaboration.

Keywords: Collaboration network · Gender · Academic network

1 Introduction

The study of how research is conducted is important in understanding the information and conclusions we accept from that research. The demographics of the researchers themselves can impact what is being studied, how the data is interpreted, and the impact of the research. One of the demographic facets that has potential affects on research collaborations is gender.

One effect of the genders of the researchers on the output is the *Matilda Effect*, named after American feminist, Matilda J. Gage. The *Matilda Effect* refers to the tendency of research produced by women to receive less attention [11]. This effect suggests that there exists a historical tendency in the academic community to be less trusting toward women researchers. A study conducted in 2006 suggested that a diversity of perspectives can provide more innovative solutions to multifaceted problems [13]. This suggests that the inclusion of researchers of different backgrounds can be beneficial to the output of studies. Applying this to gender, collaborations between researchers of different genders may be more successful. Miyoko O. Watanabe, the Deputy Executive

Director at the Office for Diversity and Inclusion at the Japan Science and Technology Agency has said that younger people are more “data-literate”, meaning they rely much more on “evidence-based data” rather than experiences to make decisions for the future [5]. For this reason, it is important to produce data analyses showing gender collaboration patterns in research in order to further encourage future diversity and inclusion measures.

We investigate the collaboration patterns between researchers of different genders at California Polytechnic State University, San Luis Obispo (Cal Poly), a largely undergraduate university with approximately 1,400 faculty members. We scope the study to the Biology, Computer Science and Software Engineering, Electrical Engineering, and Mathematics departments. These departments were chosen to compare the gender collaborations in departments with both more applied and more theoretical fields. We evaluate multiple recently published claims of gender patterns in research across these departments.

2 Related Works

A **collaboration network** is an instance of a network where vertices represent researchers and an edge exists between two vertices if the corresponding researchers collaborated on a research activity. For this paper, an edge will exist between two researchers if they collaborated on a publication.

This paper will examine collaboration patterns among researchers with respect to gender. GLAAD, an American organization dedicated to countering discrimination against the LGBTQ population, defines **gender identity** as “a person’s internal, deeply held sense of their gender”. A gender identity can be a binary gender (man or woman) or a non-binary/genderqueer. This differs from a person’s **sex** which is the classification of a person as male or female based on their external anatomy and other bodily characteristics [2].

In 2020, Elsevier, a leading information publishing and analytics company, released their third gender report, titled *The Researcher Journey Through a Gender Lens*, examining researchers in Argentina, Brazil, Mexico, Canada, USA, UK, Portugal, Spain, France, Italy, Netherlands, Germany, Denmark, Australia, and Japan as well as EU28 which aggregates data from all 28 countries in the EU. These researchers were also grouped by subject area, including disciplines within the physical sciences, life sciences, health sciences, and social sciences. This report summarizes findings from studies conducted during recent years [5]. The study found that there has been a trend toward gender parity when comparing active authors (authoring at least 2 publications during the study period) in 1999–2003 to those in 2014–2018. For every subject area in each of the countries analyzed, it was found that the median ratio of women to men among active authors was higher in the later period of 2014–2018 than in 1999–2003. Despite this trend toward gender parity, Elsevier’s study still found evidence for a gender gap. They found that even during the period from 2014–2018, most disciplines had more active male authors than active female authors. This gap was most pronounced in the physical sciences, including mathematics, computer science, and engineering disciplines [5].

In 2013, Abramo, D’Angelo, and Murgia studied gender collaboration patterns for university researchers in Italy. It was found that women’s propensity for collaboration in general was slightly higher than that of men. After separating the collaborations into intramural and extramural, they found that while across all disciplines except civil engineering, women had a higher propensity for intramural collaboration, men had a higher propensity for domestic and international extramural collaboration in mathematics, computer sciences, and biology [7].

Gender homophily occurs when male researchers collaborate more with other men and female researchers collaborate more with other women [8]. Holman and Morandin examined this pattern in the field of life sciences. They measured a gender homophily score for journals on PubMed, a biomedical article database from 2005–2016. When comparing the scores for journals from 2005–2006 and 2015–2016, they found gender homophily is greater in the recent journals than in those from the earlier period [8]. This pattern was also examined in collaborations authored by researchers in Brazil by E. Araújo, N. Araújo, Moreira, Herrmann, and Andrade. They found that while men tend to collaborate with other men, women collaborate more equally across the genders in all fields examined except for engineering [6].

In 2019, a collaboration network was built for Cal Poly researchers. This study compared the gender collaboration patterns over three nested networks: the Computer Science department network, College of Engineering network, and University-Wide network [10]. With this network, claims about gender collaboration patterns were assessed. This included measuring collaborations with each gender in total for each researcher as well as how gender collaboration patterns changed over time [10]. While the assessments of these claims has contributed greatly to understanding the state of gender collaboration patterns at Cal Poly, full verification was only conducted on the Computer Science department network, so the rest of the data was not as granularly checked [10]. In this paper, we construct a similar network for four verified departments.

3 Methods

We build a collaboration network for the Computer Science, Electrical Engineering, Mathematics, and Biology departments at Cal Poly. In this network, vertices are defined as researchers: either active faculty members at Cal Poly (as of the 2019–2020 academic year) in one of the departments of interest or a direct collaborator of one of these faculty members (either internal to Cal Poly in another department or external to the university). Two researchers are connected if they have ever coauthored a publication within the years from 1972 to 2020. Note that there is no overlap in the faculty within these four departments.

The network was primarily filled through online databases and supplemented with publication lists provided by researchers. The databases used were the Microsoft Academic Knowledge API for all departments, MathSciNet for the Mathematics department, and IEEE Xplore API for the Computer Science and Electrical Engineering department.

Microsoft’s Project Academic Knowledge exposes information from the Microsoft Academic Graph. This graph contains data mined from the Bing web index and knowledge base. This data includes scholarly activity entities including field of study, authors, institutions, and papers [4]. MathSciNet is a searchable database of mathematical publications reviewed by *Mathematical Reviews* [3]. IEEE Xplore provides access to the content published by researchers with the Institute of Electrical and Electronic Engineers (IEEE). Its content includes books, conferences, courses, and journals in the fields of electrical engineering, computer science, and electronics [1].

In order to integrate these data sources we used the above APIs on each of the names of faculty scraped from each department’s website, with the exception of MathSciNet which we manually searched. The data sources were integrated by name. Individual authors were disambiguated as follows: results were limited by profiles of authors associated with Cal Poly. For authors with many (more than 20) publications, manual checks were done by paper.

Gender Inference. Since the genders of the researchers in our network are not labeled, an API was used to label each of the researchers’ genders. We chose to use Gender API because a study by Santamaría and Mihaljevic found it to be the most accurate for gender inferences [12].

Since we were already using names as identifiers, no parsing was needed to extract names. Papers found by manual scraping had authors separated. The first names were sent to GenderAPI after being lowercased. Gender API handled the normalization. Gender API offered a threshold of confidence for each name. We took names that were above the 50% threshold. So if name X was labeled female with 51% confidence by Gender API, we took the name to be female.

We note that the use of this API is a weakness of this study. Some of our data sources only supplied first initials for the authors in each publication, which will be labeled as “unknown”. Additionally, we recognize that researchers may not identify as a binary gender, which is not considered by the Gender API and therefore could not be considered in this study.

4 Properties of the Collaboration Network

Find the vertex and edge distribution of the network in Table 1. From the Gender API, each researcher was marked as male, female, or unknown. In Table 2 are the counts of each gender in the entire collaboration network and within each of the department networks.

This data was used to analyze the validity of the following claims for researchers in the Biology, Computer Science, Electrical Engineering, and Mathematics departments at Cal Poly. In these analyses, researchers whose genders were marked as unknown were ignored.

Note that in each of the box and whisker plots, the median is represented by the yellow line and the green triangle represents the mean.

Table 1. Researcher and publication counts of the collaboration network

Department	Number of researchers (vertices)	Number of cal poly researchers	Number of publications	Number of co-authorship experiences (edges)
Entire Network	6432	158	3376	14631
Biology	1880	43	669	3615
Computer Science	1802	44	872	3786
Electrical Engineering	2387	39	1375	6000
Mathematics	453	33	475	1335

Table 2. Gender counts in the collaboration network

Department	Male	CP Male	Female	CP female	Unknown	CP unknown
Entire Network	4007	109	1489	47	936	3
Biology	1049	19	689	24	142	0
Computer Science	1312	35	354	9	136	0
Electrical Engineering	1424	35	362	4	601	0
Mathematics	285	20	106	10	62	3

5 Claims

In the following we consider four previously published observations about the difference in collaboration patterns between male and female researchers. We encourage the reader to pay particular attention to the Math department’s collaboration patterns in each of the following studies.

5.1 Claim: Men Tend to Have More Collaborators

The 2020 Gender Report found that across all departments and regions studied, men tended to have more direct collaborators than women [5]. This was also found by Ductor, Goyal, and Prummer in their study of the collaboration patterns of the Economics department at the University of Cambridge. They found that women tended to have 23% fewer collaborators than the men in the study [9].

To test this, we measured the number of different researchers each Cal Poly researcher collaborated with and took the average across each department of interest. For example, if Researcher 1 worked on Paper 1 with Researcher 2 and Researcher 3 and worked on Paper 2 with Researcher 3 and Researcher 4, Researcher 1 will have 3 collaborators.

The results of this measurement can be seen in Table 3 and Fig. 1. From the box and whisker plot, it can be seen that the mean number of collaborators of men is greater than that for women in the Biology, Computer Science, and Electrical Engineering departments at Cal Poly. We found that in the Math department, women tend to have more collaborators than men. This is especially

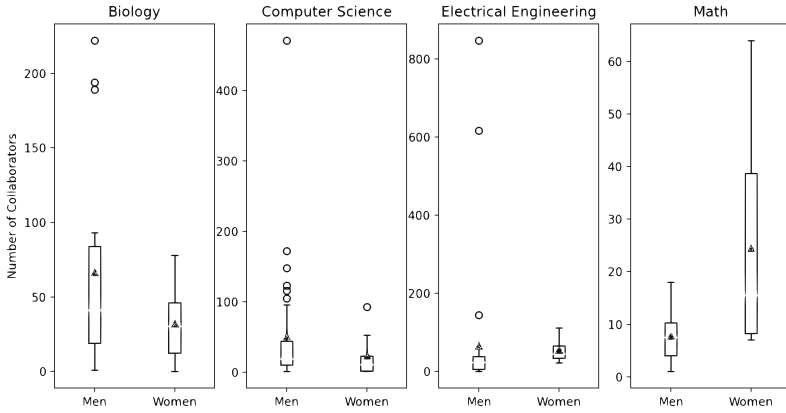


Fig. 1. Number of collaborators by department

significant because 10 out of the 30 labeled Cal Poly researchers from the Math department are women, so this represents a significant portion of the population.

Table 3. Average number of collaborators

Gender	Biology	Computer Science	Electrical Engineering	Math
Men	66.47	51.09	66.14	7.8
Women	32.13	23.67	56	24.5

5.2 Claim: Women Tend to Repeatedly Collaborate with the Same Collaborators

As part of their study of the Economics department at the University of Cambridge, Ductor, Goyal, and Prummer also found that women are more likely to collaborate with a researcher multiple times than men [9]. To measure this, they calculated a strength of ties measurement for each of the researchers, where the strength of ties of researcher i is

$$s_i = \frac{1}{d_i} \sum_j n_{ij}$$

where d_i is the number of researcher i 's distinct collaborators and n_{ij} is the number of publications written between researcher i and researcher j . They found that women had a 9.4% greater average strength of ties than men.

In order to test the presence of this pattern in the network, we measured the strength of ties for women and men in each of the departments of interest.

Table 4. Percent difference in average strength of ties

Biology	Computer Science	Electrical Engineering	Math
9.4%	-1.5%	2.5%	-50.9%

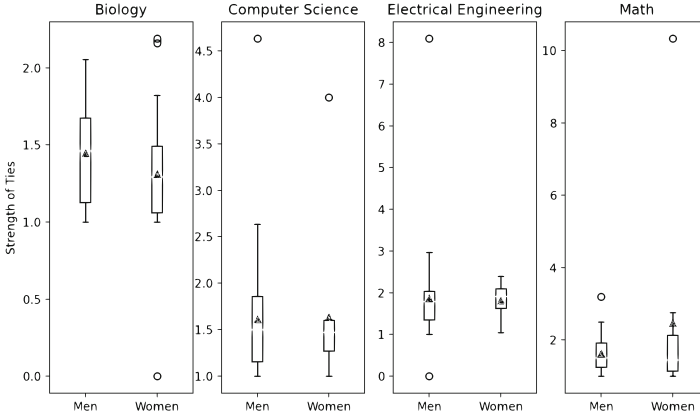


Fig. 2. Strength of ties by department

The measurements can be seen in Fig. 2. The percent differences between the men’s and women’s average strength of ties in each department can be seen in Table 4. Since the average strength of ties for women is greater than that for men in the Computer Science and Math departments, the claim that women tend to collaborate with the same collaborators is supported in those departments. In the Math department, it is important to note that there is an outlier for the strength of ties of the women, suggesting that this percent difference between the strength of ties between the men and women in Table 4 may be exaggerated. However, the opposite is true for the Biology and Electrical Engineering departments: the men tend to repeatedly collaborate with the same collaborators.

5.3 Claim: Researchers Tend to Collaborate with Authors of the Same Gender

By g-Ratio. E. Araújo, M. Araújo, Moreira, Herrmann, and Andrade found that researchers tend to collaborate with researchers of the same gender in a study conducted in Brazil [6]. They reached this conclusion by comparing g-ratios for researchers in the analyzed departments. The g-ratio for a researcher i is defined as

$$g\text{-ratio}_i = \frac{\sum_{j \in \text{women}} w_{ij}}{\sum_{j \in \text{researchers}} w_{ij}}$$

where w_{ij} is the total weight of collaborations between researcher i and researcher j . They found that women have a higher g-ratio across all fields.

The 2020 Gender Report also supported this claim using a similar measurement. The Gender Report measured the average share of woman collaborators for each of the subject areas, which is the proportion of collaborators who are women. They found that in almost all regions and subject areas, the average share of woman collaborators is greater for woman researchers [5]. Note that the average share of woman collaborators is equal to the g-ratio where the weight of all collaborations is equal to one.

To compare this finding with the data from the collaboration network for Cal Poly, we will define a weightless g-ratio, WGR since no weights were given to each collaboration. For researcher i ,

$$WGR_i = \frac{\text{Count of co-authorship experiences between researcher } i \text{ and a researcher } j \text{ who is female}}{\text{Count of total co-authorship experiences for researcher } i}$$

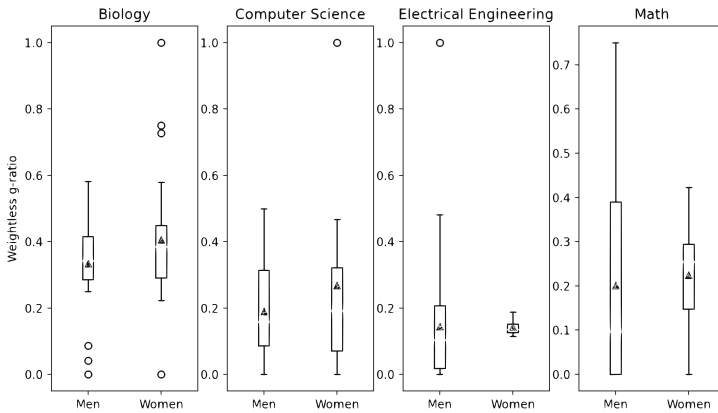


Fig. 3. Weightless g-ratio by department

Figure 3 shows the average WGRs for each department. It can be seen that in all departments except Electrical Engineering, the WGR for women is higher than that for men. This points toward gender homophily in these three departments since women show a higher ratio of co-authorship experiences with other women than men. The Electrical Engineering department shows almost equal WGR between men and women researchers, suggesting that men and women collaborate equally as often with women researchers.

By α' Metric. Evidence for gender homophily was also found to be true for researchers in the life sciences by Holman and Morandin in a study of publications from PubMed [8]. To measure gender homophily, Holman and Morandin calculated the coefficient of homophily for a collaboration network, $\alpha' = p - q$

where p is the average proportion of male co-authors on papers with a male seed author and q is the average proportion of male co-authors on papers with a female seed author. They found that for many journals, the α' was greater than 0.1, meaning that the gender ratio of men and women researchers differs on average by more than 10%.

The measured α' values for the Biology, Computer Science, Electrical Engineering, and Math departments are in Table 5.

Table 5. α' by department

Biology	Computer Science	Electrical Engineering	Math
0.00	0.10	-0.04	-0.07

While the measurements for α' for the Computer Science department seems to agree with the findings of Holman and Morandin, the measurements for the Biology, Electrical Engineering, and Math departments are low, suggesting little gender homophily. This may be attributed to the researchers labeled “unknown” or the limited number of researchers in each department. For this calculation, the α' value was calculated for each department, while Holman and Morandin calculated this value over 3308 journals to obtain this trend.

5.4 Claim: Women Tend to Collaborate More Intramurally

In a study of researchers at an Italian university, Abramo, D’Angelo, and Murgia found that women are more likely than men to collaborate intramurally, where an intramural collaboration is a collaboration with someone from the same institution [7]. These researchers defined the propensity to collaborate intramurally as $CI = cip/p$ where cip is the number of intramural collaborations and p is the total number of collaborations for the researcher. They found that on average, women had a CI of 78.9% while men had a CI of 73.9%.

To study this in our collaboration network for researchers at Cal Poly, we calculated the propensity to collaborate intramurally. This was calculated by dividing the number of “collaborator experiences” with intramural researchers by the total number of “collaborator experiences”. We define a “collaborator experience” as a paper worked on with a researcher. So, if Researcher 1 worked on Paper 1 with Researcher 2 and Researcher 3 and Paper 2 with Researcher 2, Researcher 1 would have 2 “collaborator experiences” with Researcher 2 and 1 “collaborator experience” with Researcher 3. This allows more frequent collaborators to have a greater impact on the propensity.

Figure 4 shows that in the Biology, Computer Science and Software Engineering, and Electrical Engineering departments, on average, women have a higher propensity to collaborate intramurally than men. In the Math department, the opposite holds true: men have a higher propensity to collaborate intramurally than women.

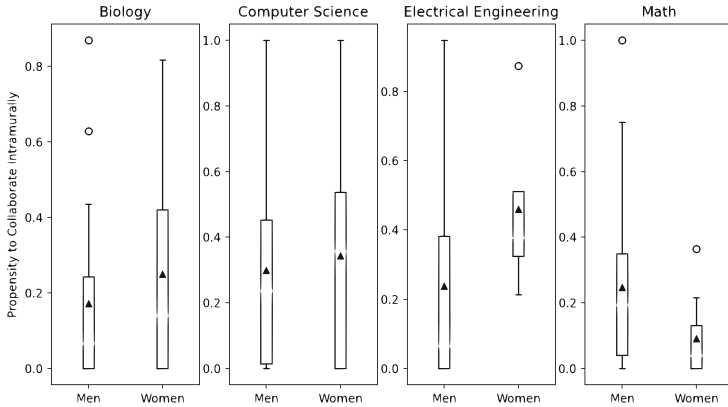


Fig. 4. Propensity to collaborate intramurally by department

6 Conclusions and Future Work

This study is ongoing with the goal of identifying departments with healthy and successful collaboration networks, especially for all genders. Our goal is to examine collaboration trends and understand how to promote equity and academic collaboration for all.

When interpreting these analyses, it is important to understand the context of these studies. These were conducted at Cal Poly, a public university in San Luis Obispo, CA. It is not a Research 1 university, meaning that while research is conducted at Cal Poly, the primary focus is on undergraduate education. This limits the amount of data that can be retrieved for each of the departments of interest because all faculty members are not required to participate in research, meaning the research output is more limited than it would be at a more research-focused institution. Nevertheless, from the findings, it can be seen that there is likely a difference in the collaboration patterns of researchers based on their gender. This difference seems to depend on the department being examined, but it does support the need for more research in this area.

A limitation of this study is that we are unable to take into account additional properties of each researcher, such as academic seniority, prestige of previous institutions, and demographic information. These factors could also have a strong influence on the claims discussed.

For many of the claims investigated in this paper, the Math department was an outlier to the other departments studied. Further study must be conducted to investigate the reasons behind these differences and whether they persist for researchers in the field of mathematics across the country or globally.

For future work we intend to investigate the collaboration patterns of the Mathematics department at Cal Poly and Mathematics departments in general. Also, we intend to develop a framework where Cal Poly researchers can self iden-

tify their gender. This would allow for a more accurate picture of collaborative patterns with respect to gender.

References

1. About IEEE Xplore. IEEE Xplore Digital Library
2. Glossary of terms - transgender. GLAAD
3. Mathscinet. American Mathematical Society
4. Project academic knowledge. Microsoft (2020)
5. The researcher journey through a gender lens. Elsevier (2020)
6. Araujo, E., Araujo, N., Moreira, A., Herrmann, H., Andrade, J.: Gender differences in scientific collaborations: women are more egalitarian than men. *PLoS ONE* **12**, 05 (2017)
7. D'Angelo, C.A., Abramo, G., Murgia, G.: Gender differences in research collaboration. *Journal of Informetr.* **7**, 811–822 (2013)
8. Holman, L., Morandin, C.: Researchers collaborate with same-gendered colleagues more often than expected across the life sciences. *PLOS ONE* **14**, e0216128 (2019)
9. Goyal, S., Ductor, L., Prummer, A.: Gender & collaboration. Cambridge Working Papers in Economics, Faculty of Economics, University of Cambridge (2018)
10. McNichols, L., Medina-Kim, G., Nguyen, V.L., Rapp, C., Migler, T.: Gender's influence on academic collaboration in a university-wide network. In: *Complex Networks and Their Applications VIII*, pp. 94–104. Springer, Cham (2020)
11. Rossiter, M.W.: The Matthew Matilda effect in science. *Soc. Stud. Sci.* **23**(2), 325–341 (1993)
12. Santamaría, L., Mihaljevic, H.: Comparison and benchmark of name-to-gender inference services. *PeerJ Comput. Sci.* **4**, e156 (2018)
13. Reich, J.A., Reich, S.M.: Cultural competence in interdisciplinary collaborations: a method for respecting diversity in research partnerships. *Am. J. Commun. Psychol.* **38**, 51–62 (2006)



Uncovering the Image Structure of Japanese TV Commercials Through a Co-occurrence Network Representation

Mariko I. Ito^(✉) and Takaaki Ohnishi

Rikkyo University, Toshima-ku, Tokyo, Japan
ito.mariko@rikkyo.ac.jp

Abstract. The effect of TV commercials on the purchase intention of the viewers has been extensively studied. The literature has suggested that some images in TV commercials positively affect the purchase intention of the viewers. However, the overall picture of the image used in TV commercials has not been sufficiently revealed. We studied the image structure of TV commercials in Japan by constructing a weighted co-occurrence network of keywords used in such advertisements. We found the cores of the image structure that frequently co-occur with other keywords in TV commercials covering various categories of products. We further conducted a community detection, where a community can be regarded as a set of keywords in particular associated with each other, based on their frequent co-occurrence in TV advertisements. The core keywords belong to different communities, and we discuss the characteristics of each community in the present paper.

Keywords: Co-occurrence network · Community structure · Image structure

1 Introduction

Companies invest significant amounts of money in TV commercials to secure the purchase intentions of customers [1, 2]. Experiments and empirical data analyses have been conducted to evaluate the actual effect of TV commercial advertisements on customer purchase intentions and behaviours. Psychological experiments have shown a phenomenon in which people tend to prefer more familiar objects, which is known as the mere-exposure effect, which supports the efficacy of TV commercial advertisements [3]. However, it has also been suggested that the mere-exposure effect can be reduced when the subjects are aware of the persuasive intent [3, 4]. Furthermore, studies have indicated that most viewers regard TV advertisements as intrusive, thereby reducing their effect [1, 4]. By contrast, for certain categories of TV advertisements, e.g., tourism and foods targeting children, studies have shown that certain images positively affect the purchase intention of the viewers [2, 5].

As a shortcoming of such previous studies, the variation in images appearing in TV commercials has not been sufficiently considered, or the categories of the TV commercials examined, as well as the types of images in them, have been limited in their examination. To properly evaluate the effect of TV advertisements, their classification according to the images included and an evaluation of the effect based on the classification is needed. However, to the best of our knowledge, an extensive investigation into how various images are used to create a TV commercial remains insufficient. In this study, we aim to demonstrate the overall picture of the image used in TV commercials in Japan.

Keyword co-occurrence networks, in which each node represents a keyword and each edge indicates the co-occurrence of two keywords found in the academic literature, have been used to investigate knowledge structures [6–9]. Following this method, we investigated the image structure of Japanese TV commercials by constructing a co-occurrence network of keywords exhibited within them. We used an immense dataset of Japanese TV advertisements, allowing us to overview the image structure of such commercials. Our analysis revealed some core keywords, e.g., ‘woman’, ‘product’, ‘logo’ and ‘man’, within the image structure. We conducted a community detection and found that these core keywords belong to different communities. Here, a community can be regarded as a set of keywords frequently co-occurring in TV commercials.

2 Methods

The dataset analysed in this study was provided by M Data Co., Ltd. (<https://mdata.tv/en/>). This dataset includes information on TV commercials aired on five TV stations in Japan, i.e. Fuji TV, Nippon TV, TBS TV, TV Asahi, and TV Tokyo, during the period of January 2017 to June 2020. A total of 1,682,171, 1,672,549, 1,657,578, and 804,828 TV commercials were recorded in 2017, 2018, 2019 and 2020, respectively. Here, commercials that have the same content but have gone to air at a different time are counted as different advertisements. In the dataset, the scenario of each TV commercial is represented by keywords, e.g., ‘nursery’, ‘pick up’, and ‘mother and child’. These keywords are from a commercial advertising Japanese dumplings and describe a situation in which a mother picks up her child from a nursery and ends up buying Japanese dumplings for dinner. Each TV commercial is classified according to the type of the product advertised. There are 38 categories in the larger classification, and all categories are further divided into a total of 131 sub-categories (Table 1).

We constructed a co-occurrence network of TV commercials, G . Firstly, for each category, we constructed an unweighted network of keywords, where each node represents a keyword and an edge exists between two nodes if these keywords appear at least once in the same TV commercial (Fig. 1). Subsequently, we merge these networks to construct a weighted network, G . Network G consists of nodes that have appeared in any of the networks of sub-categories. The weight of the edge between two nodes represents the number of networks for the sub-categories in which the edge between these nodes exists. Network G has 74,106 nodes and 1,659,254 edges. Let A be the weight matrix of G .

The strength of node i , $s_i = \sum_j A_{i,j}$, is a measure of the importance of node i in a weighted network [10,11]. Nodes with high strength can be regarded as the core of the image structure of the TV commercials because they co-occur with various keywords across a variety of sub-categories of products. We examined the strength of the nodes and the strength distribution in G .

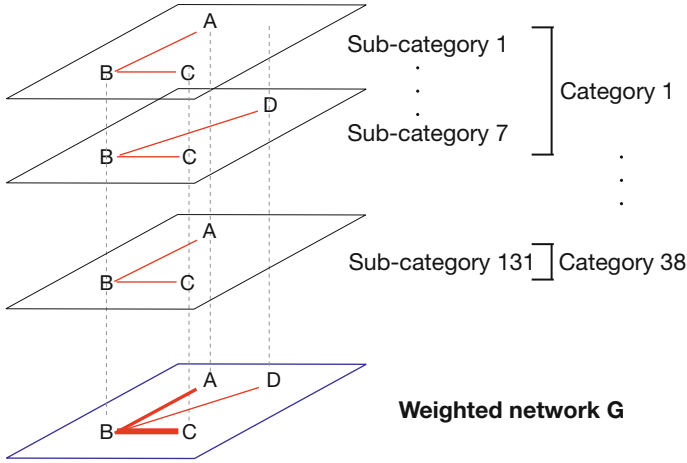


Fig. 1. Schematic image of the construction of the weighted co-occurrence network G . Nodes A, B, C, and D in this figure represent keywords in practice.

We also investigated the community structure in G . Community detection was conducted using modularity function Q :

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{s_i s_j}{2m} \right] \delta(c(i), c(j)), \tag{1}$$

where $2m$ denotes $\sum_{i,j} A_{ij}$, Kronecker’s delta is denoted by δ , and $c(i)$ indicates the community to which node i belongs [12]. Modularity Q evaluates the goodness of the graph partition. The value of $s_i s_j / (2m)$ on the right-hand side of Eq. (1) is the expected weight of the edge between nodes i and j in a random network where the strength distribution is the same as that of G . Therefore, the more two nodes assigned to the same community are connected with a large weight compared to that expected in a random network, the greater the value of Q is. We obtained a graph partition that (locally) maximises Q using Louvain heuristics [13,14]. In our case, nodes within the same community can be regarded as the set of keywords that co-occur in particular and are mutually associative in the image structure of TV commercials.

Table 1. Categories of products advertised in TV commercials, and the number of sub-categories in each category in the analysed dataset.

Category	#Sub-categories	Category	#Sub-categories
Store	7	Toy	2
Distribution industry	1	Snack	1
Apparel	3	Appliance	5
Estate	3	Household goods	5
Machine	2	Cosmetics	1
Online shopping	2	Medicine	1
Communication	8	Logistic	13
Detergent	1	Roadshow	1
Oil & Tire	1	PC & A/V	1
Draft beer	1	A/V software	1
Cup noodle	4	Canned coffee	4
Pet food	1	Drip coffee	6
Food	5	Drink	2
Wine	6	Tobacco	1
Beer	3	Sports	4
Liquor	6	Camera & Watch	3
Car	8	Interior	3
Credit card	9	Publication	2
Finance & Insurance	3	Others	1

3 Results

3.1 Degree and Strength

In the weighted co-occurrence network of keywords G , nodes with a high strength can be regarded as the cores in the image structure of TV commercials because these nodes connect to many of the other nodes over various sub-categories. Figure 2(a) shows the strength distribution of G . We can see the strong heterogeneity in the strength of the nodes, and those with extreme strength are shown in Table 2.

The degree distribution is also shown in Fig. 2(b). Weights of the edges are not incorporated into the degrees. The degree of a node stands for simply the number of other keywords that have co-occurred at least once with the node irrespective of the frequency of co-occurrences.

Nodes with a significant degree tend to also have a significant strength (Fig. 2(c)). However, the orders of the nodes according to the strength and degree are slightly different from each other. For example, ‘woman’ has the highest strength, whereas ‘product’ has the highest degree. Thus, we can infer that the co-occurrence of ‘woman’ and other keywords can be observed in TV commer-

cials focusing on more various types of products compared to that of ‘product’. For example, the weight of the edge between ‘woman’ and ‘logo’ is 111, and thus these keywords co-occur in 111 sub-categories.

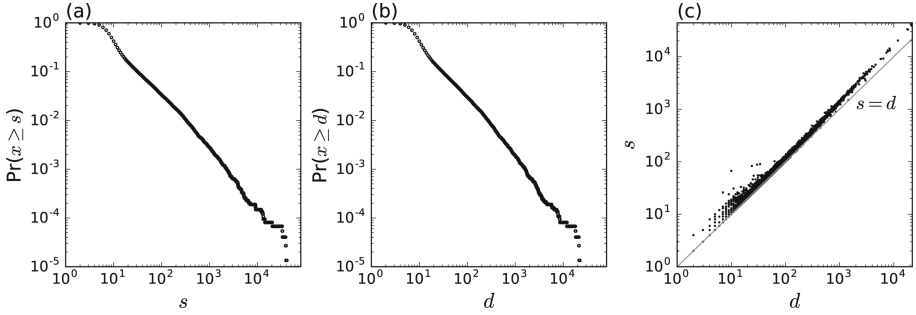


Fig. 2. Complementary cumulative distribution function (CCDF) of (a) strength and (b) degree. (c) The relationship between strength and degree of the nodes.

3.2 Community Structure

Subsequently, we show the community structure in G . Modularity Q of the resulted graph partition, which was conducted as mentioned in Sect. 2, was 0.268. Ten communities composed of a significantly large number of nodes are shown in Table 3. The eleventh community consists of 179 nodes, which is quite a small number compared to that of the first through the tenth communities. Nodes in the eleventh and lower communities may significantly reflect the contents in only a few commercials, and thus we discuss only the first to the tenth communities in this study.

The nodes in a community can be regarded as a set of nodes that are mutually associated across different sub-categories because they co-occur with each other in TV commercials throughout these various sub-categories. Six nodes with the highest strength, i.e. ‘woman’, ‘product’, ‘logo’, ‘man’, ‘white back’, and ‘cinema scope’ were assigned to different communities (Table 3). Each community seems to have a specific image. For example, nodes in community 2 tend to be related to storylines, community 5 seems to consist of nodes associated with eating and drinking, and there are many keywords related to school life in community 10. Community 7 seems to share keywords related to the basic elements in the images. The node with the greatest strength in each community can be the representative keyword of the image, e.g. ‘woman’ exhibits the greatest strength in community 9.

Table 2. Fifteen nodes with the highest strength. The degree of each node is also shown.

Node	Strength	Degree	Node	Strength	Degree
'woman'	40,850	21,167	'animation'	13,136	8,522
'product'	39,393	22,110	'black back'	12,692	7,809
'logo'	38,675	21,408	'eat'	11,935	7,320
'man'	33,220	17,869	'smartphone'	9,203	6,431
'white back'	32,744	18,223	'white back & logo'	9,071	5,820
'cinema scope'	20,306	11,932	'drink'	9,042	5,711
'characters'	14,276	8,359	'dance'	7,039	4,607
'man & woman'	13,183	7,727	'run'	6,568	3,967

We calculated the *within-module degree* z of each node. This measure indicates how much the node connects to other nodes within its community [15, 16]. For node i , the within-module degree z_i is defined as

$$z_i = \frac{s_{i,l} - \bar{s}_l}{\sigma_l}, \quad (2)$$

where community l is the one to which node i belongs, and $s_{i,l}$ denotes the sum of the weight of edges connecting node i and nodes in community l . The mean and the standard deviation of $s_{j,l}$, where node j is in community l , are denoted by \bar{s}_l and σ_l , respectively. The within-module degree z is the z-score of $s_{i,l}$ in community l , by definition. In Table 3, the star beside the node indicates that the value of z exceeds 2.5 [15]. Most nodes in Table 3 have significantly large values of z , and can be regarded as the hubs within their communities.

We further investigated the features of each community by evaluating the extent to which the community is related to each of the 38 categories of products. A node, i.e., a keyword, can appear in TV commercials in multiple categories or sub-categories. Firstly, for each category k , we count the number of sub-categories in which node i appears, $N_{i,k}$, as shown in Fig. 1. For example, the maximum value of $N_{i,k}$ is 7 when there are 7 sub-categories in category k . Subsequently, we normalise $N_{i,k}$ to compare categories, each of which has different numbers of sub-categories and nodes, as follows:

$$n_{i,k} = N_{i,k} / \sum_j n_{j,k}, \quad (3)$$

which represents the extent to which node i is related to category k . For each community l and each category k , we calculate the sum of $n_{i,k}$ of nodes that belong to community l :

$$W_{l,k} = \sum_{\text{node } i \in \text{community } l} n_{i,k}. \quad (4)$$

Let $w_{l,k}$ be the normalisation of $W_{l,k}$: $w_{l,k} = W_{l,k} / \sum_j W_{l,j}$. Therefore, $w_{l,k}$ can represent the extent to which community l is composed of nodes associated with category k . The value of $w_{l,k}$ for each community and each category is shown in Fig. 3 using a heatmap.

Table 3. Nodes in each community. Ten nodes with the highest strength are shown, and the node with the ultimate highest strength for each community is highlighted in bold. The star beside the node indicates that its within-module degree z exceeds 2.5.

Community	#Nodes	Examples of nodes
1	10,484	‘product’ *, ‘man & woman’*, ‘illustration’*, ‘surprised’*, ‘a comment’*, ‘round frame’*, ‘studio’*, ‘product in hand’*, ‘conversation’*, ‘description’*
2	9,335	‘animation’ *, ‘girl’*, ‘mascot character’*, ‘pose’*, ‘boy’* ‘call out’*, ‘title’*, ‘game’*, ‘card’*, ‘explosion’*
3	9,000	‘cinema scope’ *, ‘white back & logo’*, ‘run’*, ‘walk’*, ‘aerial view’*, ‘face’*, ‘sea’*, ‘in a car’*, ‘night’*, ‘car’*
4	8,652	‘logo’ *, ‘black back’*, ‘dance’*, ‘sing’*, ‘many people’* ‘jump’*, ‘rejoice’*, ‘live’*, ‘stage’*, ‘audience’*
5	7,611	‘eat’ *, ‘drink’*, ‘photo’*, ‘cooking’*, ‘inside store’* ‘cheers’*, ‘kitchen’*, ‘pour’*, ‘billboard’*, ‘dining table’*
6	7,257	‘man’ *, ‘smartphone’*, ‘laugh’*, ‘talk’*, ‘office’* ‘speak’*, ‘suited man’*, ‘another man’*, ‘telop’*, ‘smartphone screen’*
7	6,686	‘white back’ *, ‘characters’*, ‘red back’*, ‘blue back’*, ‘pink back’*, ‘yellow back’*, ‘yellow back’*, ‘cover’*, ‘open’*, ‘pointing’
8	5,782	‘smile’ *, ‘indoor’*, ‘family’*, ‘parent & children’*, ‘married couple’*, ‘children’*, ‘shake hands’*, ‘dog’*, ‘cat’*, ‘girl’*
9	5,700	‘woman’ *, ‘turn around’*, ‘look back’*, ‘room’*, ‘cafe’* ‘sit down’*, ‘bed’*, ‘outdoor’*, ‘turns’*, ‘paint’*
10	2,110	‘classroom’ *, ‘rooftop’*, ‘schoolgirl’*, ‘white back’*, ‘female student’*, ‘schoolboy’*, ‘corridor’*, ‘study’*, ‘confession’*, ‘views’

A single category, Publication category, is dominant in community 7, which includes ‘white back’, ‘characters’, and ‘red back’. Similarly, the Communication category is salient in community 10, including ‘classroom’, ‘rooftop’, and ‘schoolgirl’. Therefore, many keywords that frequently appear in Publication or Communication categories should belong to community 7 or 10, respectively. This may suggest that the image the nodes in community 7 (10) share is significantly used to advertise the products in the Publication (Communication) category.

Each of the other communities seems to be associated with multiple categories. For example, community 9 consists of nodes that are associated specifically with the categories Apparel, Pet food, Cosmetics, and Interior. In other words, TV commercials of the products in these categories can share the same

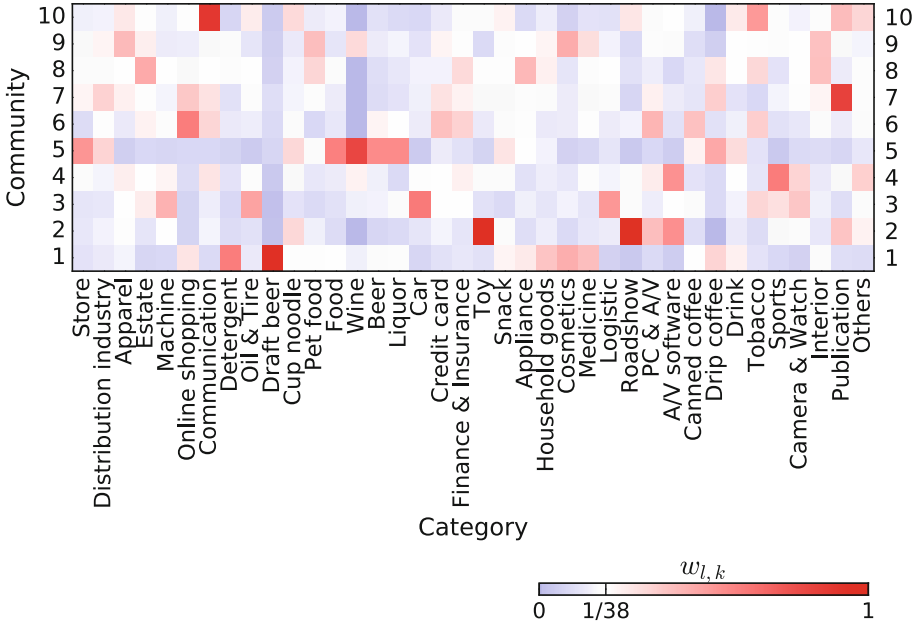


Fig. 3. Features of each community. The values of $w_{l,k}$ are shown by a heatmap for community l (the vertical axis) and category k (the horizontal axis). The average of $w_{l,k}$ is $1/38$ because there are 38 categories, and red (blue) in the heatmap indicates that the value of $w_{l,k}$ is greater (less) than the average.

image, which are represented by ‘women’, ‘turn around’, ‘look back’, ‘room’, and ‘cafe’, among others.

As another example of the relationship between communities and categories, community 1 is related to the categories Detergent, Draft beer, Household goods, Cosmetics, and Medicine, and Detergent and Draft beer, in particular, are salient only in community 1. In community 1, there are nodes associated with the product itself and its description, e.g., ‘product’, ‘product in hand’, and ‘description’. Therefore, it seems that community 1, at least partially, possesses an image of persuasion. Thus, products in the categories associated with community 1, Detergent, Draft beer, Household goods, Cosmetics, and Medicine, can be advertised directly without concealing the intended persuasion.

4 Discussion

We studied the image structure of TV commercials in Japan by constructing a weighted co-occurrence network of keywords in such advertisements. We evaluated the strength of the nodes and investigated the community structure in the co-occurrence network. Nodes with a significant strength can be regarded as the cores in the image structure [6]. A community is composed of nodes that are

mutually connected with significant strength, i.e., that are significantly associated with each other by frequently co-occurring in TV commercials across the categories of products. The cores of the image structure, ‘woman’, ‘product’, ‘logo’, ‘man’, ‘white back’, and ‘cinema scope’ were assigned to the different communities. These can be representative keywords of these communities.

The literature has shown that community detection based on the modularity optimisation has a shortcoming, called resolution limit. The resolution limit is a problem that we cannot detect communities with the relatively small size [12]. This problem comes from the assumption of the null model incorporated into modularity. In the null model, the random network with the same size as the focal network is considered, i.e., we assume that a node can interact with all nodes. Communities with the small size can be obtained by narrowing the range of possible interaction for a node in the assumption of the null model [12, 17]. This assumption may be suitable for the case of social networks, considering people’s capacity to interact with others. By contrast, there seems to be no plausible reason to narrow the range of connection when we are interested in the overall picture of the image in TV commercials. In addition, in the case of the co-occurrence network of TV commercials, communities with the small size can significantly reflect the contents in the commercials of a single product; a product is sometimes advertised by the series of commercials, that share similar story and contents. Thus, we think that a partition of the co-occurrence network by the modularity optimisation is rather preferable to reveal the image structure.

Each community seems to have a specific image; in community 2, for example, nodes are associated with storylines. As mentioned in Sect. 3, some nodes with significant strength in community 1 evoke the persuasive intent by the advertisers. The literature has suggested that the conspicuous placement of products or apparent persuasion in an advertising can induce a defensive mindset of the consumers [3, 4]. Therefore, it is presumably possible that TV commercials in the categories of Detergent, Draft beer, Household goods, Cosmetics, and Medicine, which are related particularly to community 1, do not positively affect the purchase intention of the viewers.

Our results suggest that the effect of TV commercials should be evaluated by considering the image structure of such advertisements, which are composed of multiple subsets (communities), each of which has a specific image. For example, we may have to classify TV commercials according to their image before evaluating their effect. This will lead to an understanding of which type of image can strongly influence the effect of TV commercials on the purchase intention or behaviour of the viewers.

We analysed all TV commercials in Japan during the period of 2017 to 2020. It should be interesting to study whether the nature of the image structure shown in this paper is universal in various countries or is unique to Japan. For example, community 6, which includes ‘man’, ‘smartphone’, ‘laugh’, ‘talk’, and ‘office’ among others, is associated in particular with the categories of Online shopping, Credit cards, PC & A/V, Canned coffee, and Tobacco. Whether such a mutual association of keywords in community 6 can be observed and whether these

categories of products tend to share a similar image characterised by community 6 presumably depend on the culture.

As another future perspective, investigating the temporal change of a co-occurrence network should be an interesting topic. The previous studies on knowledge structure revealed the transition of the trend or the important theme in academic fields by examining the cores (nodes with significant strength) and their surroundings. A similar analysis can be applied to the image structure of TV commercials, and may reveal the history of not only improvements in TV commercials but also the cultural transition occurring in Japan.

Acknowledgement. This work was supported by JSPS KAKENHI Grant Number JP20K19929.

References

1. Carreón, E.C.A., Nonaka, H., Hentona, A., Yamashiro, H.: Measuring the influence of mere exposure effect of TV commercial adverts on purchase behavior based on machine learning prediction models. *Inf. Process. Manag.* **56**(4), 1339–1355 (2019)
2. Boyland, E.J., Halford, J.C.: Television advertising and branding. Effects on eating behaviour and food preferences in children. *Appetite* **62**, 236–241 (2013)
3. Hekkert, P., Thurgood, C., Whitfield, T.A.: The mere exposure effect for consumer products as a consequence of existing familiarity and controlled exposure. *Acta Psychol.* **144**(2), 411–417 (2013)
4. Davtyan, D., Cunningham, I.: An investigation of brand placement effects on brand attitudes and purchase intentions: brand placements versus TV commercials. *J. Bus. Res.* **70**, 160–167 (2017)
5. Pan, S.: The role of TV commercial visuals in forming memorable and impressive destination images. *J. Travel Res.* **50**(2), 171–185 (2011)
6. Radhakrishnan, S., Erbis, S., Isaacs, J.A., Kamarthi, S.: Novel keyword co-occurrence network-based methods to foster systematic reviews of scientific literature. *PLoS ONE* **12**(3), e0172778 (2017)
7. Su, H.N., Lee, P.C.: Mapping knowledge structure by keyword co-occurrence: a first look at journal papers in technology foresight. *Scientometrics* **85**(1), 65–79 (2010)
8. Zhang, J., Xie, J., Hou, W., Tu, X., Xu, J., Song, F., Wang, Z., Lu, Z.: Mapping the knowledge structure of research on patient adherence: knowledge domain visualization based co-word analysis and social network analysis. *PLoS ONE* **7**(4), e34497 (2012)
9. Özgür, A., Cetin, B., Bingol, H.: Co-occurrence network of Reuters news. *Int. J. Mod. Phys. C* **19**(05), 689–702 (2008)
10. Lü, L., Chen, D., Ren, X.L., Zhang, Q.M., Zhang, Y.C., Zhou, T.: Vital nodes identification in complex networks. *Phys. Rep.* **650**, 1–63 (2016)
11. Barrat, A., Barthélemy, M., Pastor-Satorras, R., Vespignani, A.: The architecture of complex weighted networks. *Proc. Natl. Acad. Sci. U.S.A.* **101**(11), 3747–3752 (2004)
12. Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**(3–5), 75–174 (2010)
13. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech.: Theory Exp.* **2008**(10), P10008 (2008)

14. Aynaud, T.: python-louvain 0.14: Louvain algorithm for community detection (2020). <https://github.com/taynaud/python-louvain>
15. Guimera, R., Amaral, L.A.N.: Functional cartography of complex metabolic networks. *Nature* **433**(7028), 895–900 (2005)
16. Rubinov, M., Sporns, O.: Complex network measures of brain connectivity: uses and interpretations. *Neuroimage* **52**(3), 1059–1069 (2010)
17. Lambiotte, R., Delvenne, J.C., Barahona, M.: Laplacian dynamics and multiscale modular structure in networks. arXiv preprint [arXiv:0812.1770](https://arxiv.org/abs/0812.1770) (2008)



Movie Script Similarity Using Multilayer Network Portrait Divergence

Majda Lafhel¹(✉), Hocine Cherifi²(✉), Benjamin Renoust³,
Mohammed El Hassouni¹, and Youssef Mourchid⁴

¹ Mohammed V University in Rabat, Rabat, Morocco
majdalafhel1@gmail.com, mohamed.elhassouni@gmail.com

² LIB, University of Burgundy, Dijon, France
hocine.cherifi@u-bourgogne.fr

³ Institute for Datability Science, Osaka University, Suita, Japan
renoust@ids.osaka-u.ac.jp

⁴ L@bisen, Yncrea Ouest, Brest, France
youssefmour@gmail.com

Abstract. This paper addresses the question of movie similarity through multilayer graph similarity measures. Recent work has shown how to construct multilayer networks using movie scripts, and how they capture different aspects of the stories. Based on this modeling, we propose to rely on the multilayer structure and compute different similarities, so we may compare movies, not from their visual content, summary, or actors, but actually from their own storyboard. We propose to do so using “portrait divergence”, which has been recently introduced to compute graph distances from summarizing graph characteristics. We illustrate our approach on the series of six Star Wars movies.

Keywords: Multilayer networks · Movies · Network portrait · Network similarity

1 Introduction

Network models have been increasingly used to support the analysis of stories [11], such as novels and plays [15, 27], famous TV series [25], news [22, 23], and movies [16, 17, 19]. However, most of these models only focus on one facet of the movie story. Indeed, usually, they capture interactions between the characters at play [11] to bring out a global picture of the story content. Other works went beyond by introducing other semantic elements such as scenes and dialogues [25]. Nevertheless, they have always captured the information in a single layer network or a bipartite graph. To enrich the representation, we have previously proposed a multilayer network model that captures key elements of the movie story [16, 17]. It encompasses the single network analysis based either on characters or scenes and proposes new topological analysis tools.

Recently, various studies are being conducted to measure the similarity of movie stories [13, 14] focusing on characters interactions. This work, aims at

quantifying similarity between movies by incorporating knowledge extracted from complementary aspects of a story, including character, dialogues, and locations. To do so, rather than relying on a single-layer network, a multilayer network extracted from the movie script is exploited. It combines semantically extracted characters, keywords, and locations, allowing to compute similarity between the corresponding networks. Starting from the previously proposed model [16, 17] the multilayer network is automatically extracted from the scripts of the movies to compare. At this point the similarity issue reduces to compute the similarities between the layers. Plenty of methods have been proposed in the literature for comparing networks [26]. Here, a recent method introduced by Bagrow *et al.* [2] is used. The so-called *Network Portrait Divergence* is based on the network portrait [3], which characterizes a network by determining its degree distribution at various distances. Experiments performed on the Star Wars series show encouraging results that match up with the perceptual analysis.

The rest of the paper is organized as follows. Section 2, summarizes the *multilayer movie script model* and its *extraction process*, together with *network portrait* and its *divergence*. Section 3 starts with introducing the data then results of the evaluation are discussed. Conclusion and discussions are reported in Sect. 4.

2 Background

2.1 Extracting Multilayer Networks from Movie Scripts

The multilayer network model [16] considers three types of entities, each forming a unique layer: *characters* C , *keywords* K , and *locations* L , together with their interactions. Intralayer interactions: $C - C$ when two characters share a conversation; $K - K$ when two keywords co-occur in a conversation; $L - L$ when two locations are in successive scenes. Interlayer interactions also exist: $C - K$ when a character pronounces a keyword; $C - L$ when a character is in a location; $K - L$ when a keyword is pronounced in a location.

A movie *script* is a semi-structured text containing all technical information concerning scenes, dialogues, and settings. It is divided into *scenes* delimited by *scene heading* that specify the physical spaces (INT or EXT), location, and the time (DAY or NIGHT).

The first step of processing consists in dividing the script into scenes based on the scenes headings. Figure 1 illustrates a scene. Locations are extracted from the scene heading. Characters are then extracted as dialogues header (in capital). Finally, keywords are extracted from dialogues by applying Latent Dirichlet Allocation (LDA) [4]. Named Entity Recognition (NER) [18] is further applied to extract character and location from the keywords.

From Fig. 1, the following links and entities can be extracted: $C - C$: ANAKIN and SHMI; $K - K$: Mom and Annie; $L - L$: TUSKEN RAIDER HUTT and GEONOSIS, then GEONOSIS and TATOOINE; $C - K$: ANAKIN and Mom; $C - L$: ANAKIN and TUSKEN RAIDER HUTT, then SHMI and TUSKEN RAIDER HUTT; $K - L$: Mom and TUSKEN RAIDER HUTT.

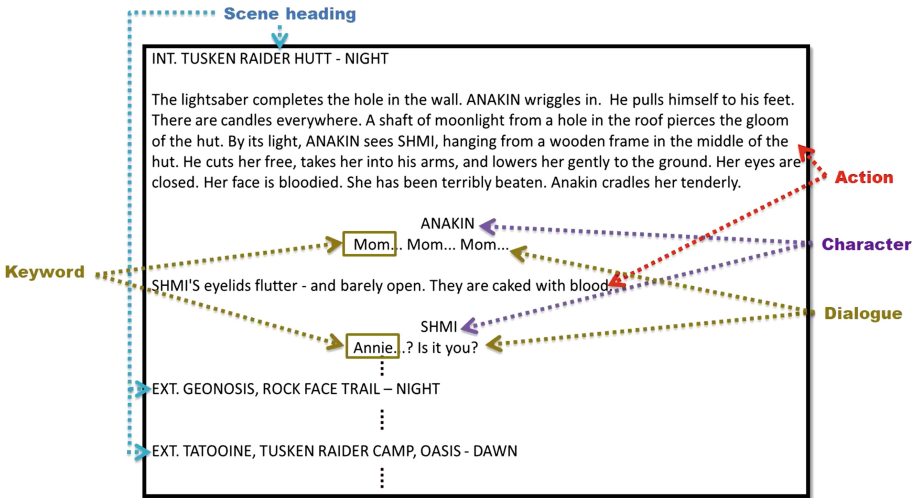


Fig. 1. Excerpt of *Attack of the Clones* script describing one scene.

2.2 Network Comparison Using Portrait and Portrait Divergence

In 2008, Bagrow introduced *Network Portraits* [3] to summarize large complex networks. The Network Portrait is a matrix B with $B_{l,k}$ elements such as defined in Eq. (1). Given a network G with N nodes, k is a number of nodes such that $0 \leq k \leq N$, and l is the shortest path length between two nodes, such that $0 \leq l \leq d$ (graph diameter).

More formally:

$$B_{l,k} \equiv \text{the number of nodes connected with } k \text{ nodes at distance } l \quad (1)$$

Each row in B represents the probability distribution $P(k|l)$ such that k nodes are reachable at distance l from a randomly chosen node:

$$P(k|l) = \frac{1}{N} B_{l,k} \quad B_{1,k} = N\mathbf{P}(k) \quad (2)$$

B is a signature of G . Bagrow then introduced a distance measure between two networks G and G' , the *Network Portrait Divergence* [2] based on this signature.

Let B and B' be the portraits associated with G and G' respectively. The portrait divergence computes first the probability distribution of nodes based on the matrices B and B' , such as in Eq. 3, 4. Then, the distance between the probability distributions is computed using the Jensen Shannon divergence D_{JS} , such as in Eq. 5.

Let G be a network of node size N . The portrait B is associated with G . Consider two randomly chosen nodes at distance l . The probability distribution $P_B(k|l)$ is defined as follows:

$$P_B(k|l) = \frac{1}{N} B_{l,k} = \frac{k B_{l,k}}{\sum_c n_c^2} \quad (3)$$

where n_c is the number of nodes inside each connected component c . Accordingly, let G' be a network with a total number of nodes N' with a portrait B' , the probability distribution $P_{B'}(k|l)$ is defined as follows:

$$P_{B'}(k|l) = \frac{1}{N'} B'_{l,k} = \frac{k B'_{l,k}}{\sum_c n_c'^2} \quad (4)$$

where n_c' is the number of nodes inside each connected component c .

The *Network Portrait Divergence* between G and G' is defined as follows:

$$D_{JS}(G, G') = \frac{1}{2} (KL(P_B || P_*) + KL(P_{B'} || P_*)) \quad (5)$$

where P_* is a combination of P_B and $P_{B'}$, such as: $P_* = \frac{(P_B + P_{B'})}{2}$, and $KL(\cdot || \cdot)$ is the Kullback-Liebler divergence. The KL divergence within two probability distributions P and P' is defined as:

$$KL(P(k|l) || P'(k|l)) = \sum_{l=0}^{max(d,d')} \sum_{k=0}^N P \log \left(\frac{P(k|l)}{P'(k|l)} \right) \quad (6)$$

3 Experimental Evaluation

Experiments are performed using the movie scripts of six episodes of the Star Wars saga¹. It is divided into the sequel (original) and prequel trilogies. The sequel trilogy is the first created by George Lucas. It is composed of Episode IV (SW4): *A New Hope* (1977), Episode V (SW5): *The Empire Strikes Back* (1980) and Episode VI (SW6): *Return of the Jedi* (1983). It is followed by the prequel trilogy composed of Episode I (SW1): *The Phantom Menace* (1999), Episode II (SW2): *Attack of the Clones* (2002), and Episode III (SW3): *Revenge*

¹ Here is a quick summary of the plot: The saga follows Anakin Skywalker, a young child freed from slavery to become a Jedi and endeavored to save the galaxy. Anakin instructed by the Jedi Masters of the light side, married the senator Padme. Unfortunately, the Sith (Palpatine) submits him to the dark side, rebelling and losing against his Master (Obi-Wan). Anakin is saved by the Sith, now ruling over the galaxy, and transformed to Darth Vader. Padme died while giving birth to twins Luke and Leia. Luke becomes a farmer while Leia becomes a princess. Nineteen years later, Obi-Wan met Luke and taught him the Jedi way, while receiving a distress call from the princess Leia, leading the resistance against Palpatine. Joining smuggler Han Solo in the Millenium Falcon they went to save her, and support the resistance. Luke completes his Jedi training, while Solo gets captured by the Sith, who crushes most of the resistance. Vader tries to turn Luke to the dark side when discovering that Luke is his son. Unsuccessful, Palpatine tries to kill Luke, awaking in Vader his old self. Vader turns back against Palpatine and rescues the galaxy.

of the *Sith* (2005). The experiment process proceeds as follows. First, the multi-layer network of each episode of the saga are extracted. Their basic topological properties are summarised in Table 1. Then, the corresponding *portrait* for each layer (*character*, *location*, and *keyword*) of each multilayer network are computed individually and discussed. Finally, the distance between each pair of same-type layers between all movies are compared using their *portrait divergence*.

Table 1. Global properties of the character, keyword and location layers for each movie, with: number of nodes (N), number of edges (E), diameter (D), transitivity (T), and assortativity (A).

Episode	Character layers					Keyword layers					Location layers				
	N	E	D	T	A	N	E	D	T	A	N	E	D	T	A
SW1	61	1090	3	0.53	-0.12	229	4492	4	0.66	-0.01	116	26	2	0	-0.03
SW2	52	506	3	0.43	-0.14	244	6009	4	0.52	-0.01	118	24	2	0	0.24
SW3	56	476	4	0.53	-0.008	233	4532	5	0.45	0.003	152	16	2	0	-0.19
SW4	58	576	4	0.51	-0.11	234	2201	7	0.61	0.11	133	428	6	0.42	0.17
SW5	45	504	5	0.48	0.15	211	568	6	0.31	0.08	145	132	4	0.12	-0.2
SW6	44	266	4	0.42	0.34	213	2648	4	0.45	-0.03	84	30	4	0.2	-0.27

3.1 Comparing Portraits

Comparing all Three Layers: A quick look at all three layers: *characters* in Fig. 2, *keywords* in Fig. 3, and *locations* in Fig. 4, clearly shows that each type of layers display a distinct pattern making it recognizable. *Character* layers display a k between 15 and 30, for a maximum path length between 3 and 5. *Keyword* layers do not display much larger path length (up to 6) but the number of nodes are much larger. Despite this larger number of nodes, most nodes are distributed in the lower path corner with degree 1 and 3, maybe suggesting a lot of small components. The *character* and *keyword* layers both display clear characteristics common to small-world and scale-free networks [3] in their shape such as the elongated form and knot near the center of the portrait. It may be less obvious for the *character* layer, but one should remember that character graphs are rather small. *Location* layers diverge the most from small world patterns and seem to greatly vary upon trilogies, while being minimalist in the prequel series. This suggests that locations networks are probably more linear loops in the prequel while they show more complexity in the original trilogy.

Comparing Character Layers: Each character layer portrait is illustrated in Fig. 2. The overall six portraits display a very similar pattern. SW1 (Fig. 2(a)) and SW2 (Fig. 2(b)) – a surprising similarity given that this layer shows a double number of edges as shown in Table 1 – together with SW5 (Fig. 2(e)) and SW6 (Fig. 2(f)) make two similar pairs, with SW3 (Figs. 2(c)) and SW4 (Figs. 2(d)) being somewhat intermediary between them. In SW5 and SW6, a lot of action

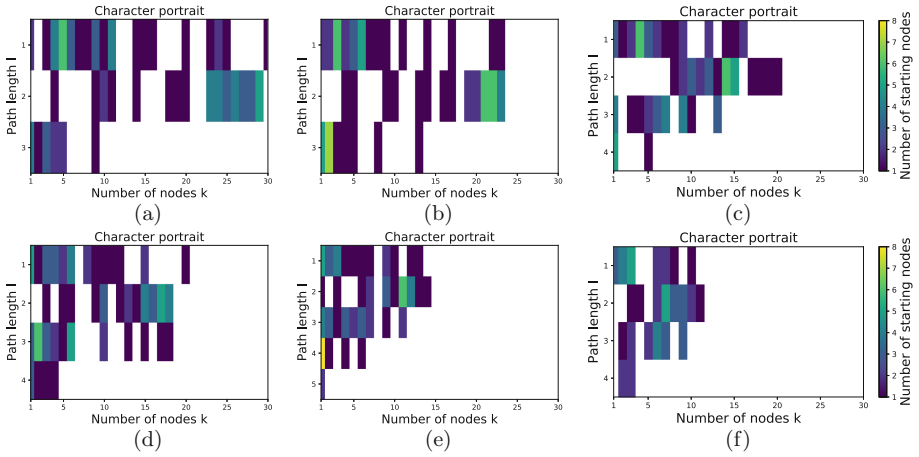


Fig. 2. Portraits of the characters layer for each movie of the Star Wars saga. (a) SW1, (b) SW2, (c) SW3, (d) SW4, (e) SW5, (f) SW6. The horizontal axis is the node degree k . The vertical axis is the distance l . Colors are the entries of the portrait matrix B_{lk} . The white color indicates $B_{lk} = 0$.

separates the main characters into different groups with parallel actions, while the first series is rather focused on the main character, Anakin. The aspect ratio of the character layer portraits seems to vary upon the different movies. This has been linked to the small world characteristics of the graph that each portrait summarizes [3]. We can notice a progression from SW1 to SW6 suggesting a “smaller” world effect for episodes 5 and 6.

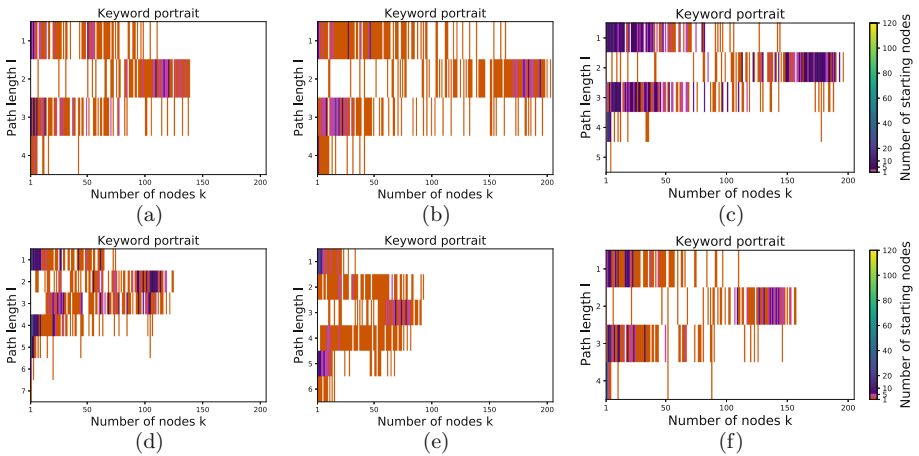


Fig. 3. Portraits of the keyword layers for each movie of the Star Wars saga. (a) SW1, (b) SW2, (c) SW3, (d) SW4, (e) SW5, (f) SW6. The horizontal axis is the node degree k . The vertical axis is the distance l . Colors are the entries of the portrait matrix B_{lk} . The white color indicates $B_{lk} = 0$.

Comparing *Keyword* Layers: Each keyword layer portrait is reported in Fig. 3. These layers can be well distinguished from the character layers portrait. Nevertheless, they also exhibit similar patterns across episodes. The aspect ratio of the keyword layer portraits also seems to vary upon the different movies, but not as regularly as the character layer. Especially, the first (Fig. 3(a)) and last (Fig. 3(f)) episodes seem to share a very similar pattern. SW2 (Fig. 3(b)) and SW3 (Fig. 3(c)) also display a close pattern with an elongated aspect ratio, while SW4 (Fig. 3(d)) and SW5 (Fig. 3(e)) portraits are much shorter. Interestingly, the portrait of this latter one differs in its lowest path length, but all portraits present the “knot” characteristic to scale-free networks. This visual proximity between keyword layers can be expected due to the tendency of language-based graphs to follow Zipf’s law [9].

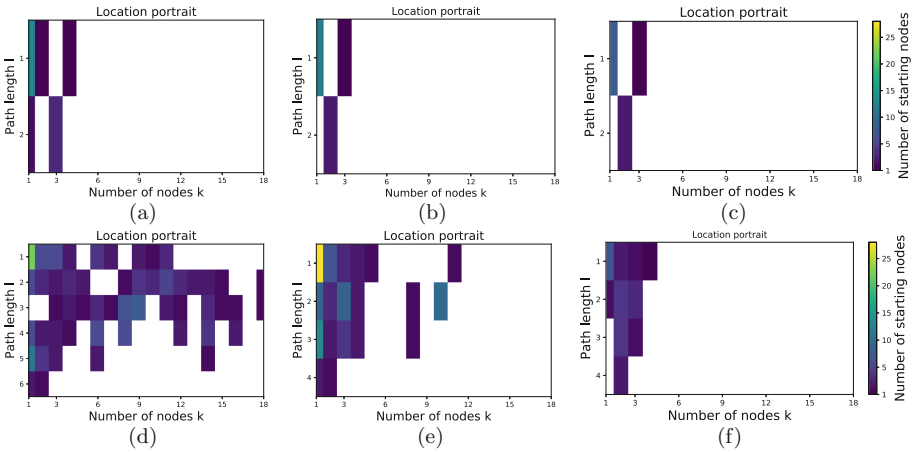


Fig. 4. Portraits of the locations layer for each movie of the Star Wars saga. (a) SW1, (b) SW2, (c) SW3, (d) SW4, (e) SW5, (f) SW6. The horizontal axis is the node degree k . The vertical axis is the distance l . Colors are the entries of the portrait matrix B_{lk} . The white color indicates $B_{lk} = 0$.

Comparing *Location* Layers: Each location layer portrait is illustrated in Fig. 4. At a glance, one can recognize a pattern very different to the character and keyword layers. The pattern is common to all three episodes of the prequel trilogy (Fig. 4(a–c)). It shows very low maximum path length and degree. This strongly suggest the presence of chain-type topology. As a consequence, the cut of these three movies between locations may be quite linear. Indeed this trilogy mostly follows the actions of Anakin and Obi-wan, very often together. The three episodes of the original trilogy display a more complex structure. The structure is more complex in the first (Fig. 4(d)) of these episodes, then gradually simplify to the last episode (Fig. 4(f)). In SW4, iconic star ship locations are introduced, such as the Millenium Falcon and the Death Star, and the movie cuts a lot scenes between star ships and other locations (this can be confirmed from the

number of edges in Table 1). The rhythm is also different in the original trilogy, which often separates its main characters such that actions happen in parallel. Frequent cuts between them generate transitions that are less linear. This trend seems to be less adopted after the fourth episode.

3.2 Comparing Portrait Divergence

Figure 5 illustrates the *network portraits divergence* between each SW movie. Figure 5(a) shows the divergence between character layers, Fig. 5(b) shows divergence between keyword layers, and Fig. 5(c) shows the divergence between location layers.

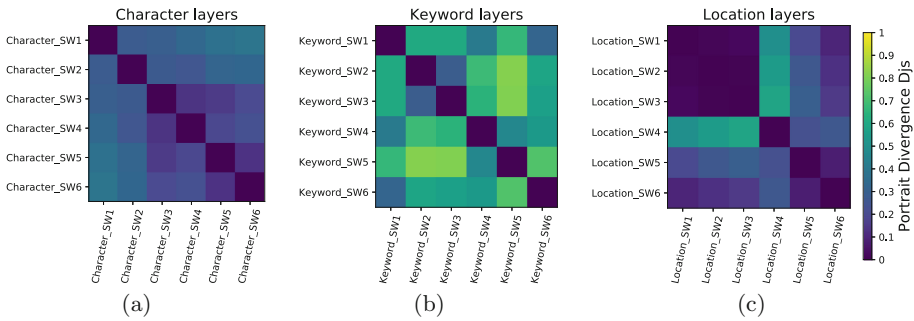


Fig. 5. Portrait divergence of the *characters*, *keywords*, and *locations* networks in the 6 episodes of the Star Wars saga. (a) *Character* layers. (b) *Keyword* layers. (c) *Location* layers. Each cell of the heat-map represents the portrait divergence between a couple of episodes.

Overall Comparison: Comparing overall layer divergence suggests that relationships in character layers are very similar along the saga. This confirms the intuition we can have at a glance from the portraits figures. Location layers also show an overall high similarity. The keyword heat-map departs from the others. Indeed, differences are more marked between the keyword layers that exhibit a more patchy structure. This is surprising considering that the visualization of each portrait made one movie to another looks very similar. It is however harder to make sense of why the fifth episode stands out. Finally, the separation between each trilogy appears clearly in the location layer. Note that the pairwise character layer similarity also makes sense considering the movies.

Divergence Between *Characters* Layers: From Fig. 5(a), the prequel trilogy (SW1–3) shows a low divergence between characters. Relationships between characters of SW1 appear more closely related to SW2 ($D_{JS} = 0.279$) than to SW3 ($D_{JS} = 0.299$). That may be due the appearance and disappearance of some key characters. For instance, Shmi dies in SW2. In SW3, Anakin changes side to become Vader, and the clones start to appear. This last episode shows an even

lower divergence with the sequel: the clones being instrumental in the original series may help shaping a common structure of relationships. This may also confirm the efforts taken in the third episode of the prequel to link characters with the original series.

In the original trilogy, SW5 and SW6 show the lowest divergence ($D_{JS} = 0.132$). Indeed, individual inspection shows that a lot of the links are shared in both episodes. SW4 appears the most diverging with the other movies in the trilogy. This makes sense since the movie was made so that it could have been a stand alone movie at first. It shows its closest relationship with SW3 ($D_{JS} = 0.134$), once more underlining the special care taken to connect the prequel to the sequel from the character relationships point of view. SW1 and SW6 show the highest divergence. This is not surprising since at this point the story of the prequel and original trilogies are completely different in their characters and plots.

Divergence Between *Keyword* Layers: The keyword layers show the highest divergence among the different movies (Fig. 5(b)). In both trilogies, the divergence appears lower for a pair of movies, SW2 and SW3 in the prequel ($D_{JS} = 0.29$), and SW4 and SW5 in the sequel. SW1 shows its lower divergence with SW4 ($D_{JS} = 0.4$) and SW6 ($D_{JS} = 0.33$). It remains difficult to conclude what in the structure of the movies can lead to a similar relationship structure in keywords between those three. We suspect the frequent association of the key terms *young* and *master* in these movies to play some important role.

Divergence Between *Location* Layers: Observing Fig. 5(c), the portrait divergence shows very high similarity within the structure of locations of the first prequel. The original series also shows a low divergence between movies. We observe a noticeable difference between the prequel trilogy and SW4, specifically, between SW3 and SW4 with $D_{JS} = 0.585$. From SW4, a lot of scenes start taking place in the Millennium Falcon, and death stars, giving a specific rhythm in cuts and locations. Nonetheless, divergence is rather low between the prequel and the last two episodes of the series. Remembering that all information is extracted from the scripts, we may suspect here different influences the movie director may have. Indeed, SW4 was written to possibly be a standalone movie, while the following SW5 and SW6 scripts were written in a short period of span, planned to complete the trilogy. This concerns even more the prequel trilogy which was planned from the beginning to be a unified trilogy.

4 Discussion and Conclusion

In this work, a multilayer network model is used to represent the main elements of a movie script: characters, key elements of the conversations (keywords), and locations of the scenes. Investigations based on the single-layer components of the model are performed to relate the similarity of the 6 Star Wars Saga episodes to the distance between the layers based on portrait divergence.

Preliminary results using portrait and portrait divergence measures are quite encouraging. Indeed, the analysis of the movie networks confirms a good correspondence between characters of the prequel and sequel trilogies; and a difference between locations in the prequel and sequel trilogy. Each of the prequel and sequel trilogies shows a good relationship between characters; and high similarity between locations. Results show similarity between topic relationships (keywords) in the movies SW2 and SW3; and also of SW1 with SW4 and SW6. Otherwise, other episodes appear to be dissimilar, mainly, the relation connecting keywords of SW5 with SW2 and SW3. Although more experiments are needed to fully assess that movie similarities can be discovered by measuring their network distances, this work opens multiple research directions.

Note that the current results are only based on the sole script of each movie. The script of SW4 was written in the late 70's, the two following movies in the 80's and the prequel in the late 90's/2000's. How movies are scripted have definitely changed over this span, and may impact our results. To even go further, we can enrich the model. Previous works have also formulated multi-media cues for movie analysis [17] including its visual features often important for recommendation systems [8, 12, 20, 28]. We may further enrich the model, using for example sentiment analysis. To further evaluate how the network similarity performs, we wish to further proceed with user evaluation, and script-based topical [5] and style [10] similarity.

Our processing scheme is currently exploiting layers of the multilayer network as separated single-layer networks. One interesting task would be to compare with distance measures of multilayer networks. Since the Kullback-Liebler divergence is not a metric and does not respect the triangular inequality, although tempting, we have found irrelevant to average divergence over all layers and measure overall similarity between movies. Alternative strategies could be to project the whole multilayer network into a single-layer knowledge graph, but such projection remains unsatisfying. We further need to investigate pure multilayer approaches and include in the comparison the effect of transition layers. We can try embedding-based comparisons [1], and investigate proper multilayer metrics such as entanglement [24] or nodes and edge coupling [6] between movies. This however requires to align named entities between layers and movies, such that they refer to the same entity. This is not straightforward since we have at least two entities with an ambiguous definition which are Anakin/Vader and Amidala/Padme. Finally, we believe this methodology to be useful once incorporated into recommendation systems to increase their efficiency [21] since we provide a different definition of a genre based on actual elements of the movie content [7].

References

1. Bagavathi, A., Krishnan, S.: Multi-net: scalable multilayer network embeddings. arXiv preprint [arXiv:1805.10172](https://arxiv.org/abs/1805.10172) (2018)
2. Bagrow, J.P., Bollt, E.M.: An information-theoretic, all-scales approach to comparing networks. *Appl. Netw. Sci.* 4(1), 45 (2019)

3. Bagrow, J.P., Bollt, E.M., Skufca, J.D., Ben-Avraham, D.: Portraits of complex networks. *EPL (Europhys. Lett.)* **81**(6), 68004 (2008)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *J. Mach. Learn. Res.* **3**(Jan), 993–1022 (2003)
5. Bougiatiotis, K., Giannakopoulos, T.: Content representation and similarity of movies based on topic extraction from subtitles. In: *Proceedings of the 9th Hellenic Conference on Artificial Intelligence*, pp. 1–7 (2016)
6. Cozzo, E., Kivelä, M., De Domenico, M., Solé-Ribalta, A., Arenas, A., Gómez, S., Porter, M.A., Moreno, Y.: Structure of triadic relations in multiplex networks. *New J. Phys.* **17**(7), 073029 (2015)
7. Deldjoo, Y., Schedl, M., Elahi, M.: Movie genome recommender: a novel recommender system based on multimedia content. In: *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*, pp. 1–4. IEEE (2019)
8. Demirkesen, C., Cherifi, H.: A comparison of multiclass SVM methods for real world natural scenes. In: *International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 752–763. Springer, Heidelberg (2008)
9. Grabska-Gradzińska, I., Kulig, A., Kwapien, J., Drożdż, S.: Complex network analysis of literary and scientific texts. *Int. J. Mod. Phys. C* **23**(07), 1250051 (2012)
10. Kovacs, B., Kleinbaum, A.M.: Language-style similarity and social networks. *Psychol. Sci.* **31**(2), 202–213 (2020)
11. Labatut, V., Bost, X.: Extraction and analysis of fictional character networks: a survey. *ACM Comput. Surv. (CSUR)* **52**(5), 1–40 (2019)
12. Lasfar, A., Mouline, S., Aboutajdine, D., Cherifi, H.: Content-based retrieval in fractal coded image databases. In: *Proceedings 15th International Conference on Pattern Recognition, ICPR-2000*, vol. 1, pp. 1031–1034. IEEE (2000)
13. Lee, O.J., Jo, N., Jung, J.J.: Measuring character-based story similarity by analyzing movie scripts. In: *Text2Story@ ECIR*, pp. 41–45 (2018)
14. Lee, O.J., Jung, J.J.: Explainable movie recommendation systems by using story-based similarity. In: *IUI Workshops* (2018)
15. Markovič, R., Gosak, M., Perc, M., Marhl, M., Grubelnik, V.: Applying network theory to fables: complexity in Slovene Belles-Lettres for different age groups. *J. Complex Netw.* **7**(1), 114–127 (2018)
16. Mouchid, Y., Renoust, B., Cherifi, H., El Hassouni, M.: Multilayer network model of movie script. In: *International Conference on Complex Networks and their Applications*, pp. 782–796. Springer (2018)
17. Mouchid, Y., Renoust, B., Roupin, O., Văn, L., Cherifi, H., El Hassouni, M.: Movienet: a movie multilayer network model using visual and textual semantic cues. *Appl. Netw. Sci.* **4**(1), 1–37 (2019)
18. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. *Linguisticae Investigationes* **30**(1), 3–26 (2007)
19. Park, S.B., Oh, K.J., Jo, G.S.: Social network analysis in a movie using character-net. *Multimed. Tools Appl.* **59**(2), 601–627 (2012)
20. Pastrana-Vidal, R., Gicquel, J., Blin, J., Cherifi, H.: Predicting subjective video quality from separated spatial and temporal assessment. In: *Human Vision and Electronic Imaging XI*, vol. 6057, p. 60570S. SPIE (2006)
21. Reddy, S., Nalluri, S., Kuniseti, S., Ashok, S., Venkatesh, B.: Content-based movie recommendation system using genre correlation. In: *Smart Intelligent Computing and Applications*, pp. 391–397. Springer (2019)
22. Renoust, B., Kobayashi, T., Ngo, T.D., Le, D.D., Satoh, S.: When face-tracking meets social networks: a story of politics in news videos. *Appl. Netw. Sci.* **1**(1), 4 (2016)

23. Renoust, B., Le, D.D., Satoh, S.: Visual analytics of political networks from face-tracking of news video. *IEEE Trans. Multimed.* **18**(11), 2184–2195 (2016)
24. Škrlj, B., Renoust, B.: Patterns of multiplex layer entanglement across real and synthetic networks. In: *International Conference on Complex Networks and Their Applications*, pp. 671–683. Springer (2019)
25. Tan, M.S., Ujum, E.A., Ratnavelu, K.: A character network study of two sci-fi tv series. In: *AIP Conference Proceedings*, vol. 1588, pp. 246–251. AIP (2014)
26. Tantardini, M., Ieva, F., Tajoli, L., Piccardi, C.: Comparing methods for comparing networks. *Sci. Rep.* **9**(1), 17557 (2019). <https://doi.org/10.1038/s41598-019-53708-y>
27. Waumans, M.C., Nicodème, T., Bersini, H.: Topology analysis of social networks extracted from literature. *PLoS ONE* **10**(6), e0126470 (2015)
28. Zhou, H., Hermans, T., Karandikar, A.V., Rehg, J.M.: Movie genre classification via scene categorization. In: *Proceedings of the 18th ACM International Conference on Multimedia*, pp. 747–750 (2010)



Interaction of Structure and Information on Tor

Mahdieh Zabihimayvan¹(✉), Reza Sadeghi², Dipesh Kadariya³, and Derek Doran³

¹ Department of Computer Science, Central Connecticut State University,
New Britain, CT, USA

Zabihimayvan@ccsu.edu

² Department of Electrical and Computer Engineering and Computer Science,
University of New Haven, West Haven, CT, USA

Rsadeghi@newhaven.edu

³ Department of Computer Science and Engineering, Wright State University, Dayton, OH, USA

{Kadariya.2, Derek.Doran}@wright.edu

Abstract. Tor is the most popular dark network in the world. It provides anonymous communications using unique application layer protocols and authorization schemes. Noble uses of Tor, including as a platform for censorship circumvention, free speech, and information dissemination make it an important socio-technical system. Past studies on Tor present exclusive investigation over its information or structure. However, activities in socio-technical systems, including Tor, need to be driven by considering both structure and information. This work attempts to address the present gap in our understanding of Tor by scrutinizing the interaction between structural identity of Tor domains and their type of information. We conduct a micro-level investigation on the neighborhood structure of Tor domains using struc2vec and classify the extracted structural identities by hierarchical clustering. Our findings reveal that the structural identity of Tor services can be categorized into eight distinct groups. One group belongs to only Dream market services where neighborhood structure is almost fully connected and thus, robust against node removal or targeted attack. Domains with different types of services form the other clusters based on if they have links to Dream market or to the domains with low/high out-degree centrality. Results indicate that the structural identity created by linking to services with significant out-degree centrality is the dominant structural identity for Tor services.

Keywords: Tor · Dark web · Structural identity · Socio-technical networks · Struc2vec

1 Introduction

Tor is an important socio-technical system which is used as a tool for internet censorship circumvention, releasing information to the public, sensitive communication between parties, and as a private space to trade goods and services [1]. Perhaps our best understanding of Tor is limited to the present art in evaluating Tor which studies its structure or information exclusively and narrowly. However, activities in socio-technical systems,

and hence Tor, are driven by both structure and information. Thus, there is still a gap in our understanding of the interplay between the structure and information on Tor.

Structural identity is a concept of categorizing network nodes based on the structure of relationships they have with others [2]. Assuming that the structural identity of Tor domains denotes their neighborhood structure independent of their service type or their location in the network, there are open questions whether the structural identity of Tor domains has any relationship with the type of services they provide and if there is any dominant structural identity on Tor. Answering these questions will give us hints on micro-level connections each domain has in the network. It will also reveal how different the Tor domains are in their structural identity and what makes this difference. If there is any relation between the service type and the neighborhood structure of dark domains, such an insight can be useful in predicting links between a new domain with the others based on their service type. Scrutinizing the dense or sparse patterns in relationships among domains also leads to predicting the proportions of the Tor network which have vulnerability against node removal or targeted attack.

To this end, we conduct a micro-level investigation on the neighborhood structure of Tor domains using `struc2vec` and classify the extracted structural identities by hierarchical clustering to study any relationship between domains' structural identity and the type of service they provide. Our findings reveal that the structural identity of Tor services can be categorized into eight distinct groups. One group belongs to only Dream market services where neighborhood structure is almost fully connected and thus, robust against node removal or targeted attack. This insight helps track and trace moving vendors of the Dream market domains based on the linking patterns in the neighborhood structure of their services. Domains with different types of services form the other clusters based on if they have links to Dream market or to the domains with low/high out-degree centrality. Results indicate that the structural identity created by linking to services with significant out-degree centrality is the dominant structural identity for Tor services.

This paper is organized as follows: Sect. 2 discusses the related work on investigating Tor. Section 3 explains how to extract the structural identity of Tor domains and presents the evaluation results and analyses over service types of domains. Finally, Sect. 4 summarizes the main conclusions and discusses the future work.

2 Related Work

Previous research on Tor can be categorized into two classes: (1) work which has focused on topological properties of Tor network; and (2) studies on characterizing different types of information and services hosted on Tor.

The topological properties of Tor, at physical and logical levels, are only beginning to be studied. O'Keeffe et al. analyzed the hyperlink structure of Tor services and compared it with the structure of the World Wide Web. Their comparison described the dark Web as a set of largely isolated dark silos, which can reveal different social behavior of dark Web users [3]. Sanchez-Rola et al. conducted a broader structural analysis over 7,257 Tor domains [4]. They find a surprising relation between Tor and the surface Web: there are more links from Tor domains to the surface Web than to other Tor domains. Bernaschi et al. presented a characterization study on topology of Tor network graph

and investigated the persistence of hidden services and their hyperlinks [5]. All analyses are conducted over three different snapshots of Tor captured during a five-month period. They also compared Tor with other social networks and surface Web graphs using well-known metrics. In another similar work [6], Bernaschi et al. investigated measurements to evaluate and characterize Tor hidden services data and topology of their network. They provided a critical discussion on possible data collection techniques for dark Web and conducted analyses on the relationship between Tor English content and its topology. In one of our previous work [7], we presented a broad evaluation of the network of referencing from Tor to surface Web and investigated to what extent Tor hidden services are vulnerable against the information leakage caused by linking to the surface websites.

Towards understanding types of content on Tor, Dolliver et al. used geo-visualizations and exploratory spatial data analyses to analyze distributions of drugs and substances advertised on the Agora Tor marketplace [8]. Results demonstrate that drugs with European sources are randomly distributed and six countries, with Canada and the United States at the top, have the major portion of drug dealing around the world. Chen et al. sought an understanding of terrorist activities by a method incorporating information collection, analysis, and visualization techniques from 39 Jihad Tor sites [9]. An expert evaluation on the proposed method indicates its high performance in investigating terrorist activities on the dark Web. Mörch et al. analyze the nature and accessibility of information related to suicide [10] by investigating the search results of nine popular search engines on Tor. Experiments depict that in comparison with the surface Web, searching “suicide” and “suicide method” on Tor results in much smaller number of sites providing suicide-related content. Biryukov et al. investigated the content and popularity of Tor hidden services by scanning their descriptors for open ports and looking at their request rate [1]. The results indicate that the content of over four fifths of Tor hidden services is in English and near half of them are devoted to drugs, adult, counterfeit, and weapon topics.

3 Structural Identity of Tor Domains

Now, we present the analyses conducted on structural identity of Tor domains. First, we describe how to extract the structural identity of Tor domains using the struc2vec algorithm. Then, we present the evaluation results of conducting hierarchical clustering over the embedding vectors which represent the structural identity of Tor domains in our data.

In the experiments, we utilize the data collected and labeled in our previous work [11] which is the product of crawling over 1 million pages from 20,000 Tor seed addresses, yielding a collection of 7,782 English pages coming from 1,766 unique Tor domains. Figure 1 presents the distribution of Tor domains based on their service types. It illustrates that directory and shopping domains and the Dream market dominate by accounting for 58.83% of all domain types in our data. In contrast, domains of Forum, Email, and News sites account for 23.66% of all the domains and Gambling, Bitcoin, and Multimedia domains constitute the smallest proportion of the Tor domains in our data. Table 1 presents a summary description of the Tor domain types.

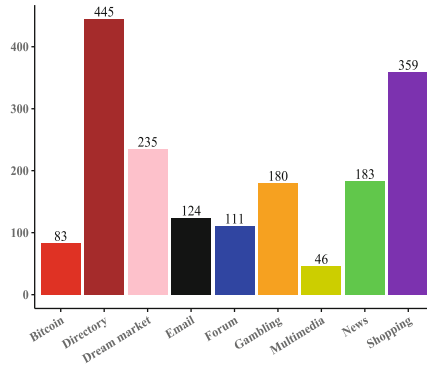


Fig. 1. Topic distribution of Tor domains

Table 1. Summary description of Tor domain types

Topic label	Description
Directory	Unnamed pages with address lists of Tor hidden services including TorDir, Hidden Wiki, and DuckDuckGo
Bitcoin	Services for Bitcoin transactions and fund transfers to wallets
News	Pages akin to personal weblogs where authors write essays on various topics and visitors post comments
Email	Communication services like email, chat room, and Tor VPNs
Multimedia	Multimedia products like academic and press articles even if they are copyright protected
Shopping	Dark markets to purchase goods, including drugs, and consultancy and investment services
Forum	Bulletin board and social network services for Tor users to discuss ideas
Gambling	Services to bet money on games, purchase gambling consulting, and read gambling-related news
Dream market	A shopping domain so large that it merited its own topic category

3.1 Representation of Tor Structural Identity

The main purpose of this work is to specify the structural identity of Tor domains based on their network structure. This identity reveals latent similarities among domains regardless of the services they provide (as vertices' labels) and their position in the Tor network. To this end, we employ the struc2vec algorithm to learn the latent representations for the structural identity of Tor domains. Struc2vec [2] is a technique to learn a language model from a network as vector embedding for vertices. The basic steps of struc2vec are as follows:

- Compute structural similarity between all node pairs: this similarity should be independent of the node or edge attribute, node position in the network, and even the

network connectivity. The latent representations of nodes should be also strongly correlated to their structural similarity. Hence, the more similar network structures of two nodes, the closer their latent representations. To do so, this step uses a hierarchical similarity metric to capture more information on the structural similarity of nodes for different neighborhood sizes.

Suppose that $G = (V, E)$ indicates the graph of G with the node set V ($n = |V|$) and the edge set E . Let d^* indicate the network diameter, and $S_d(u)$, $d \geq 0$ denote the set of nodes at distance d of the vertex u . Also, for $S \subset V$, $os(S)$ denotes the ordered degree sequence of vertices in S . The elements in S are integers in the range of $[1, n - 1]$ and repetition is possible. To impose a hierarchy to measure the latent similarities, comparisons between ordered degree sequences of each pairs of nodes is considered. If $f_d(u, v)$ denotes the similarity between u and v based on their d - and (less than d)-hop neighbors (including all the edges among them), it is defined as follows:

$$f_d(u, v) = f_{d-1}(u, v) + g(os(S_d(u)), os(S_d(v))), d \geq 0 \quad (1)$$

where $g(os(S_d(u)), os(S_d(v)))$ indicates the distance between two ordered degree sequences of $os(S_d(u))$ and $os(S_d(v))$. It is worth mentioning that $f_d(u, v)$ is only defined when both vertices have nodes at distance d . And, $f_d(u, v) = 0$ if $d \geq 0$, and $f_{d-1}(u, v) = 0$ if d -hop neighbors of both vertices are isomorphic and thus, map the both vertices onto each other. Since $S_d(u)$ and $S_d(v)$ can have different sizes, to compute their distance, Dynamic Time Warping (DTW) is used to loosely compare the patterns in sequences with different sizes [2]. To do so, DTW matches the elements of sequences in a way that sum of distances between pairs of elements will be minimized. Assuming $e_i \in S_d(u)$ and $e_j \in S_d(v)$, their distance, $dist(e_i, e_j)$ is computed as follows:

$$dist(e_i, e_j) = \frac{\max(e_i, e_j)}{\min(e_i, e_j)} - 1 \quad (2)$$

– Generate a weighted multilayer graph to encode the structural similarities between vertices: suppose that M indicates such a graph, it is comprised of $d^* + 1$ layers where each layer $d = 0, \dots, d^*$ is a weighted undirected complete graph of $|V|$ nodes and n^2 edges, and it is defined by the d -hop neighbors of nodes. Edge weights are also defined as the inverse proportion of the similarity calculated for their corresponding node pairs. If $wt_d(u, v)$ indicates weight of the edge between u and v in the layer d , it is defined as follows:

$$wt_d(u, v) = e^{-f_d(u, v)} \quad (3)$$

Based on the definition, edges corresponding to the vertices with high similarity to a vertex will have large weights across all the layers. To connect the layers, each vertex in layer d is attached to its corresponding vertex in the layers $d - 1$ and $d + 1$ using directed edges. The weights of such edges are defined as below:

$$wt(E_{u_d, u_{d+1}}) = \log(\Omega_d(u) + e), d \in [0, d^* - 1] \quad (4)$$

$$wt(E_{u_d, u_{d-1}}) = 1, d \in [1, d^*]$$

where $\Omega_d(u)$ is as follows:

$$\Omega_d(u) = \sum_{v \in V} I(\text{wt}(E_{u_d, v_d}) > \overline{\text{wt}_d}) \quad (5)$$

$I(\text{st})$ indicates an indicator function, returning 1 if st is true. Based on the definitions, $\Omega_d(u)$ shows the number of edges of vertex u with weights larger than the average edge weight in the layer d ($\overline{\text{wt}_d}$). In other words, it represents the similarity of u to other nodes in the layer d .

– Produce structural context for nodes: this step uses a biased random walk moving on the multilayer graph M to generate node sequences. Assuming that the probability of staying at the current layer is $Pr_s > 0$, the probability of moving from node u to node $v \in V(u \neq v)$ in the layer d is defined as:

$$Pr_d(u, v) = \frac{e^{-f_d(u, v)}}{\sum e^{-f_k(u, v)}} \quad (6)$$

According to the definition of $Pr_d(u, v)$, the probability of moving to nodes which are structurally more similar to u will be higher than the probability of moving to nodes with small similarity.

Given the probability of stepping to another layer, $1 - Pr_s$, the random walk will move to the layers $d + 1$ or $d - 1$ based on the following probabilities:

$$Pr_d(u_d, u_{d+1}) = \frac{\text{wt}(E_{u_d, u_{d+1}})}{\text{wt}(E_{u_d, u_{d+1}}) + \text{wt}(E_{u_d, u_{d-1}})}$$

$$Pr_d(u_d, u_{d-1}) = 1 - Pr_d(u_d, u_{d+1}) \quad (7)$$

Every vertex that the random walk moves to in a layer will be added to the context. This process of random walking is started for each node from layer 0 and repeated for a certain number of times.

– Learn the language model: this step generates the language models of node sequences created in the previous step using Skip-Gram [12]. Main purpose of Skip-Gram is to maximize the likelihood of each vertex's context in the corresponding sequence.

3.2 Clustering Tor Structural Identity

We classify the set of vectors using hierarchical clustering to see how vectors that belong to domains with similar service type locate in the same clusters. To ensure that the clustering method is able to find small clusters, we utilize Agglomerative Hierarchical clustering algorithm (AGNES) which has a bottom-up approach in generating clusters and is able to capture clusters with small size [13]. To define the similarity between two clusters, we employ the Ward's minimum variance method [14] which minimizes the final within-cluster variance by merging the pair of clusters that have minimum sum of squared Euclidean distance.

The reason of choosing the hierarchical clustering is that it avoids the problem of choosing the number of clusters before running the algorithm [15]. Dendrogram which captures the results of hierarchical clustering provides a visualization of the clusters at different granularities. This visualization can help determine the number of clusters with no need to rerun the algorithm. Hierarchical clustering also allows to cope with more intricate shapes of clusters in contrast to some methods such as Kmeans or mixture models with more restrictive assumptions on data [15].

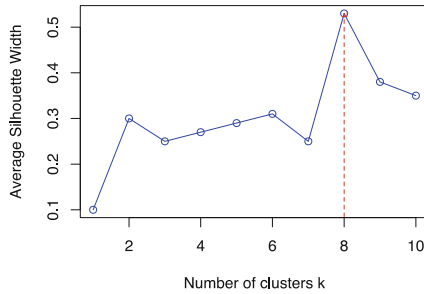


Fig. 2. Average Silhouette width vs. different number of clusters

In this work to avoid any bias towards our a priori knowledge, we leverage average Silhouette width [16] to specify the number of clusters. This metric is a graphical aid to evaluate clustering validity based on comparison of similarity among clusters. In other words, it measures how cohort a sample is with other samples in the cluster. Ranging in $[-1, +1]$, Silhouette width equal to $+1$ for a sample denotes that it is perfectly matched to its own cluster and poorly matched to others. The higher the average Silhouette width, the more appropriate the clustering configuration. Figure 2 illustrates the values of this metric for different number of clusters from 1 to 20. As indicated, the average Silhouette width for eight clusters has the maximum value (0.54). Hence, in the following analysis we consider eight clusters for the original data.

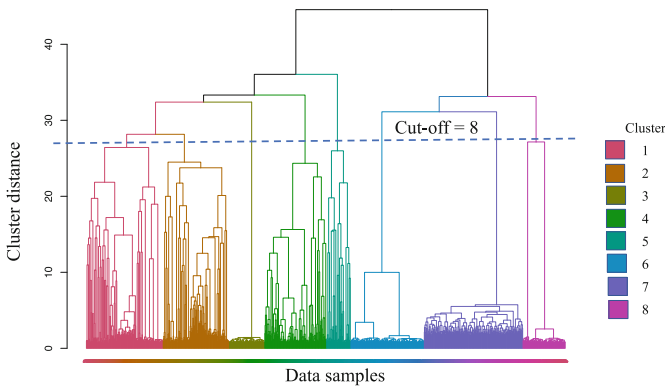


Fig. 3. Dendrogram of the Hierarchical Clustering

Figure 3 indicates the dendrogram of the hierarchies found in the data. With regard to cut-off equal to 8 as the number of clusters, first five clusters have a hierarchy separated from clusters 6, 7, and 8 with distance (dissimilarity) larger than 40. The distance between clusters 1 and 2 is less than 30, which is lower than the distance between other neighboring clusters. Clusters 1, 2, and 7 have the largest size among the others while clusters 3, 5, and 8 have the smallest size. To find a better insight into the clustering results, we further investigate the type of services for domains in each cluster (Fig. 4¹).

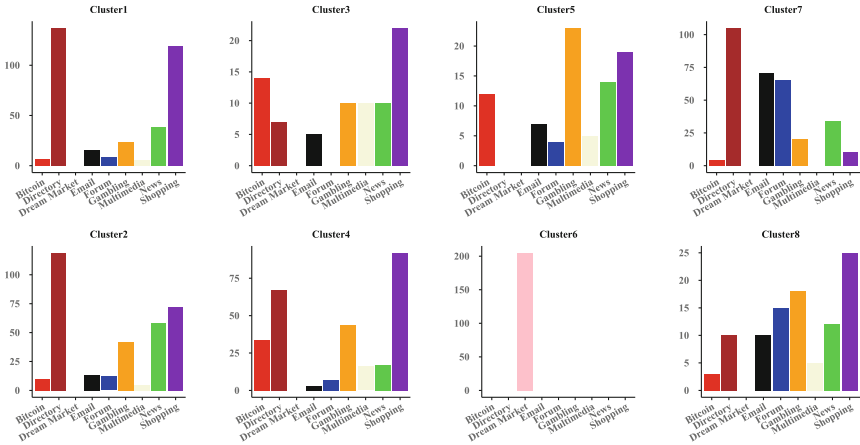


Fig. 4. Label Distributions in the resulted clusters

As Fig. 4 illustrates, one group is belonged to Dream market while the others contain different types of services. As reported in [11], the Dream market domains have a tendency towards isolation which makes them separated from the other domains in the Tor network. This is in compliance with our finding which reveals the Dream market services have a distinguished structural identity. It also implies that with high probability, a new Dream market service contains hyperlinks to other Dream market domains which are active on the dark Web. Manual exploration over the cluster 6 reveals that neighborhood structure of the Dream market services is almost fully connected, which makes their intra-connectivity robust against node removal or targeted attack. This is also in accordance with our previous finding in [11] which reported the high modularity and dense intra-connectivity of the Dream marketplaces. Figure 5 denotes some neighborhood structures of these services in the data. The vertices with maximum size indicate the domains whose neighborhood structure is extracted.

Further investigation over samples in clusters 1 to 5 shows that by average, more than 80% of samples in these clusters belong to domains that have no link to Dream market in their 1st- and 2nd- degree neighborhood. Directories and/or shopping domains are the majority in clusters 1 to 4. Manual investigation within these clusters indicates that their samples belong to domains which have links to domains with high out-degree

¹ As mentioned before, samples within a cluster represent the structural identity of the Tor domains, and the labels of samples indicate the service type provided by the related domain.

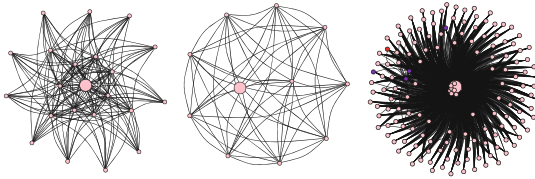


Fig. 5. Examples of neighborhood structure for Dream market domains

Table 2. Basic statistics of the out-degree centrality for Tor domains

Statistic	Value
Min	0.000
Max	407.000
Mean	6.486
Median	2.000
First quartile	1.000
Third quartile	5.000

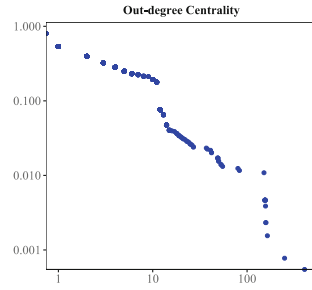


Fig. 6. CDF plot of out-degree centrality

centrality. To have better insight into such clusters, we consider basic statistics of out-degree centrality, presented in Table 2, and its CDF plot, illustrated in Fig. 6.

As the Table 2 presents, the distribution of out-degree centralities is right-skewed since the mean is larger than the median and the median is closer to the first quartile rather than the third one. The CDF plot also denotes the probability of values larger than the average notably decreases from 1 to lower than 0.1. In fact, only 19% (249 domains) of Tor domains studied in this work have out-degree centralities larger than the average.

The domains belonging to the clusters 1 to 4 have direct links to the domains with out-degree centralities greater than the average and this linking makes their 2nd-degree neighborhood large. On the other hand, gambling is the major population of domains in cluster 5. Investigation shows that the out-degree centrality of 73% of the domains in cluster 5 is lower than the average (Table 2) and they link to other domains which also have out-degrees centralities lower than this value. Therefore, their 1st-degree neighborhood is small, sparse, and thus, vulnerable against node removal.

Regarding the cluster sizes, we observe that the first four clusters contain 60% of the Tor domains studied in this work. As mentioned, these clusters belong to domains which have links to services with high out-degree centrality. The 1st-degree neighborhood of the domains can be small or large depending on their out-degree centrality. Figure 7 illustrates some examples of 1st-degree neighborhood structure of such domains (specified with larger size).

Figure 8 demonstrates some examples of the 2nd-degree neighborhood structure of the domains in clusters 1 to 4. As illustrated, due to linking to services with high out-degree centrality, the 2nd-degree neighborhood graph is large, dense, and hence, robust

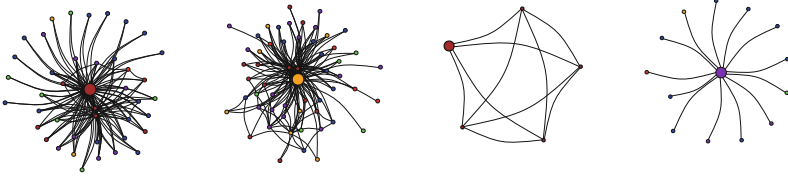


Fig. 7. Examples of 1st-degree neighborhood structure in clusters 1 to 4 (The Figs. 7, 8, and 9 are best viewed digitally and in color.)

against node removal. This implies that in spite of the sparse nature of Tor domains reported in [4, 11], the structural identity of more than half of them indicates a 2nd-degree neighborhood which is robust against node removal.

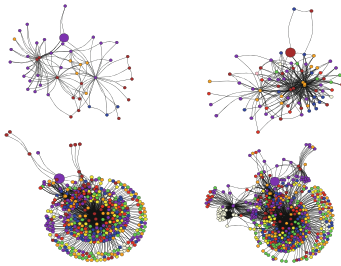


Fig. 8. Examples of 2nd-degree neighborhood structure in clusters 1 to 4

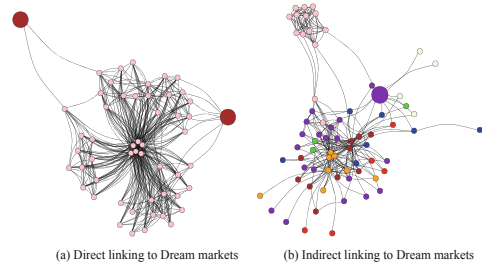


Fig. 9. Examples of neighborhood structure in clusters 7 and 8

Further investigation over the samples in clusters 7 and 8 reveals that in contrast to the first five clusters, 88.4% of domains in cluster 7 have direct links to Dream market while 92% of domains in the cluster 8 have Dream market in their 2nd-degree neighborhood (indirect linking). This explains the reason of having two distinct hierarchies in Fig. 3 between clusters 1 to 5 and clusters 6 to 8. Therefore, not only Dream market has its own structural identity, its existence in the 1st- or 2nd-degree neighborhood of domains can make different structural identities for the Tor domains. As Fig. 9(a) indicates, two directories have direct links to Dream market which directly effects on their 1st-degree neighborhood. Based on Fig. 4, directory has the maximum size in cluster 7, which is in accordance with the reports in [11] which reveals that the directories have the largest number of hyperlinks to Dream market. On the other hand, Fig. 9(b) shows one shopping domain (represented by the largest vertex) which has indirect linking to the Dream market domains. Regarding the number of shopping domains in different clusters, our analysis indicates that although shopping domains have the major population in cluster 8 (25%), this portion belongs to only 7% of all the shopping services in our data. This indicates that there is only a small number of shopping domains which have Dream market in their 2nd-degree neighborhood. However, this implies that Dream market, as the competitor of the dark shopping services, is only 2-hop far from them which can make it more difficult for the shopping service owners to attract and keep their customers' attention.

4 Conclusion and Future Work

This paper presents our investigation over structural identity of Tor domains and attempts to answer whether there is any relationship between structural identity of Tor domains and the type of information and services they provide. To this end, we employed struc2vec to extract the structural identity of Tor domains independently of their service type or their location in Tor. We utilized hierarchical clustering to classify the structural identities of domains and investigate any relationship the identities have with the domains' service types. Based on our results, the structural identity of Tor services can be categorized into eight different groups. Structural identity of the Dream market domains makes them distinct from others: an almost fully connected structure which makes the neighborhood structure robust against node removal. Domains with direct or indirect linking to the Dream market domains have structural identities which are different with the other domains. The structural identity created by linking to services with high out-degree is the dominant structural identity for Tor services, which makes their 2nd-degree neighborhood robust against node removal or targeted attack. In contrast, few domains have small sparse neighborhood structures which are vulnerable against node removal.

As the future work, more comprehensive investigations will be done to verify our experiments on the structural identity of Tor domains for datasets that differ in terms of size and method of data collection. We will also compare other classification methods including deep neural network to investigate if there is any other classifier which can outperform the hierarchical clustering used in this study. Considering graph-based clustering algorithms, e.g. the cut clustering algorithm [17], designed to detect neighborhoods/communities in graphs is another direction to extend this work.

References

1. Biryukov, A., Pustogarov, I., Thill, F.: Content and popularity analysis of tor hidden services. In: 34th International Conference on Distributed Computing Systems Workshops (2014)
2. Ribeiro, L.F., Saverese, P.H., Figueiredo, D.R.: struc2vec: learning node representations from structural identity. In: The 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2017)
3. O'Keeffe, K.P., Griffith, V., Xu, Y., Santi, P., Ratti, C.: The darkweb: a social network anomaly. arXiv preprint [arXiv:2005.14023](https://arxiv.org/abs/2005.14023) (2020)
4. Sanchez-Rola, I., Balzarotti, D., Santos, I.: The onions have eyes: a comprehensive structure and privacy analysis of tor hidden services. In: The 26th International Conference on World Wide Web (2017)
5. Bernaschi, M., Celestini, A., Guarino, S.: Spiders like onions: on the network of tor hidden services. In: The World Wide Web Conference (2019)
6. Bernaschi, M., Celestini, A., Guarino, S., Lombardi, F.: Exploring and analyzing the Tor hidden services graph. *ACM Trans. Web* **11**(4), 1–26 (2017)
7. Zabihimayvan, M., Doran, D.: A first look at references from the dark to surface web world. arXiv preprint [arXiv:1911.07814](https://arxiv.org/abs/1911.07814) (2019)
8. Dolliver, D.S., Ericson, S.P., Love, K.L.: A geographic analysis of drug trafficking patterns on the tor network. *Geogr. Rev.* **108**(1), 45–68 (2018)
9. Chen, H., Chung, W., Qin, J., Reid, E., Sageman, M., Weimann, G.: Uncovering the dark web: a case study of jihad on the web. *J. Am. Soc. Inform. Sci. Technol.* **59**(8), 1347–1359 (2008)

10. Mörch, C.-M., Louis-Philippe, C., Corthésy-Blondin, L., Plourde-Léveillé, L., Dargis, L., Mishara, B.L.: The darknet and suicide. *J. Affect. Disord.* **241**, 127–132 (2018)
11. Zabihimayvan, M., Sadeghi, R., Doran, D., Allahyari, M.: A broad evaluation of the tor english content ecosystem. In: *The 10th ACM Conference on Web Science* (2019)
12. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *Advances in Neural Information Processing Systems* (2013)
13. Rokach, L., Maimon, O.: Clustering methods. In: *Data Mining and Knowledge Discovery Handbook* (2005)
14. Ward Jr., J.H.: Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.* **58**(301), 236–244 (1963)
15. Murtagh, F.: A survey of recent advances in hierarchical clustering algorithms. *Comput. J.* **26**(4), 354–359 (1983)
16. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987)
17. Flake, G.W., Tarjan, R.E., Tsioutsoulouklis, K.: Graph clustering and minimum cut trees. *Internet Math.* **1**(4), 385–408 (2004)



Classifying Sleeping Beauties and Princes Using Citation Rarity

Takahiro Miura^(✉), Kimitaka Asatani, and Ichiro Sakata

Department of Technology Management for Innovation,
Graduate School of Engineering, The University of Tokyo, Bunkyo-ku, Tokyo, Japan
{miura, asatani, isakata}@ipr-ctr.t.u-tokyo.ac.jp

Abstract. The scientific community sometimes resists important scientific findings initially. This is the so-called “delayed recognition.” A “sleeping beauty (SB),” a representative phenomenon of delayed recognition, is a paper reported by a Prince (PR) paper. The SB includes many key breakthrough concepts for resolving scientific problems. Although many PRs discover their SBs, it is still unknown how they do that because the citation culture differs depending on the category of the paper. This study classifies SBs and their PR pairs using citation rarity within clusters that represent a unique category of a paper. Results show that citation rarity corresponds to the types of contributions to PR papers. Rare citations explore methodological insights into PR fields. Meanwhile, common citations can lead to rediscovery of the core concepts of a sleeping beauty. Furthermore, informatics and materials sciences cover major studies that include citations for SBs, whereas biological subjects find key papers through rediscovery. Results indicate that different categories of citations yield different types of SBs.

Keywords: Bibliometrics · Cross-disciplinary · Princes · Sleeping beauties

1 Introduction

Often, some of the innovative scientific works go unnoticed for long periods. This phenomenon is known as “delayed recognition” [1–3]. New discoveries and theories are significantly important for scientific progress; however, initially, they are often restricted or neglected as the scientific community is skeptical about them [4, 5]. Further, information explosion prevents important ideas from penetrating the wall of established wisdom related to a subject. Mechanisms underlying delayed recognition are always relevant to major scientific progress or groundbreaking scientific revolutions. However, how this delayed recognition occurs remains unknown.

The quantitative concept of delayed recognition, as proposed by Van Raan, can be designated simply as a sleeping beauty (SB) phenomenon [6]. Although a set of papers might go unnoticed for a long time, the same set will be suddenly noticed after a certain point a time. In addition to the original definition of SB

using depth, length, and waking up from sleep [6], several extended terms exist for the extraction of various cases of SB papers [7, 8].

Initially, SB was regarded as a rare phenomenon in scientific progress, but recent research shows that it is far less exceptional than previously thought. In fact, SBs include a number of scientific finding-related information [8].

Every SB has its own PR, which wakes it up and introduces it to the wider research community by citing the SB document. The first report to cite SB is the original definition of a PR [6]. However, this definition is suitable only for cases of “coma sleep,” i.e., cases wherein no attention was paid to citations [9]. The Internet makes it easy to access minor but related articles. Therefore, a co-citation criterion is appropriate for finding a PR [10].

Many studies have positioned SBs and PRs in a specific field or category [8, 11]. Nevertheless, there has been no systematic approach reported till date that can find SB–PR pairs comprehensively from articles because so many patterns show how a PR discovers an SB. While examining the computer science category specifically, it has been found that SBs contribute to some methodologies. Actually, PRs have extended the model and methodology established for SBs to make them applicable in other sub-fields [11]. Comprehensive analysis of SB–PR pair findings is essential because it remains unknown whether citation distributions for different sciences are similar.

Our research specifically examines classification of the various types of scientific findings across respective scientific disciplines using SB and PR pairs in various fields. The SB and PR pairs include breakpoints of the scientific findings in the concerned field. Comparison for a case of delayed recognition reveals cross-disciplinary similarity in the structure with respect to how delayed recognition is resolved. This might be the first report related to a study analyzing the number of SB–PR papers and categorizing their types.

The driving hypothesis of this paper is that estimation of the cross-disciplinary relation between SBs and PRs is performed through citation rarity calculated from complex citation networks. For this study, we have systematically clarified the relation between SBs and PRs by categorizing them post large-scale acquisition of SB and PR pairs. As a classification technique, we have considered the inadequacy of citation of SB by PR deduced on the basis of inter-cluster distance calculated with respect to complex networks corresponding to the citations.

2 Results

2.1 Sleeping Beauties and Princes

There are various methods to identify SBs, such as an average-based approach [6, 12], a quartile-based approach [13, 14], and a non-parametric approach. In this research, we have used the “beauty-coefficient,” which is a non-parametric method, for extracting SBs proposed by Ke [8] and, subsequently, for classifying the SB papers. This is because average-based and quartile-based approaches are strongly affected by arbitrary parameters of citation thresholds, which depend

on their categorical citation bias [15]. For specific examination of articles that have sufficient impact on the scientific community, we have extracted the top 5% citations from the Scopus comprehensive database. The number of top citation papers are 3,392,918, and the fewest citations are 67. As shown below, we calculated the beauty-coefficient score B for each paper.

$$B = \sum_{t=0}^{t_m} \frac{c_{t_m-c_0} \cdot t + c_0 - c_t}{t_m \max\{1, c_t\}} \quad (1)$$

In the above equation, c_t represents the number of citations that the paper received after its publication in the t th year, and t_m represents the year in which the paper received maximum citations c_{t_m} .

The Eq. (1) penalizes early citations as the later the citations are accumulated, the higher is the value of index B . We have defined the top 1% of the B scores as SB papers, which include 33,939 papers.

For each SB paper, a candidate for the PR paper is the one with the highest number of co-citations among all the papers citing that SB. For definition of SB papers, we have used the Ke's awakening year [8], which describes the time of citation burst as follows.

$$t_a = \arg\{\max_{t \leq t_m} d_t\} \quad (2)$$

$$d_t = \frac{|(c_{t_m} - c_0)t - t_m c_t + t_m c_0|}{\sqrt{(c_{t_m} - c_0)^2 + t_m^2}} \quad (3)$$

If the candidate paper was published within 5 years (i.e., around t_a , which is the awakening year of the SB papers), then it was defined as the PR paper of the SB. Thus, the number of SB-PR pairs was 14,317. Figure 1(a) presents the year-wise distribution of SB and PR. By definition, the greater the time distance between SB and PR, the larger the likely beauty coefficient. Therefore, most of SBs are papers published between 1970 and 1990. The gap year distribution reflects that (Fig. 1(b)) SBs are usually discovered after around 25 years.

2.2 Defining the SB-PR Pair Density

In this section, we have defined the SB-PR pair density with respect to its citation probability. We clustered the citation network of 67 million papers using the Leiden algorithm [16]. Citation probability is defined on the basis of the frequency of the edges between two clusters in the PR publication year. When papers in a cluster comprising a PR paper cite the particular cluster that includes the SB paper, the presence of edges between the SB and PR is not so unusual. Hence, the density in this case is high.

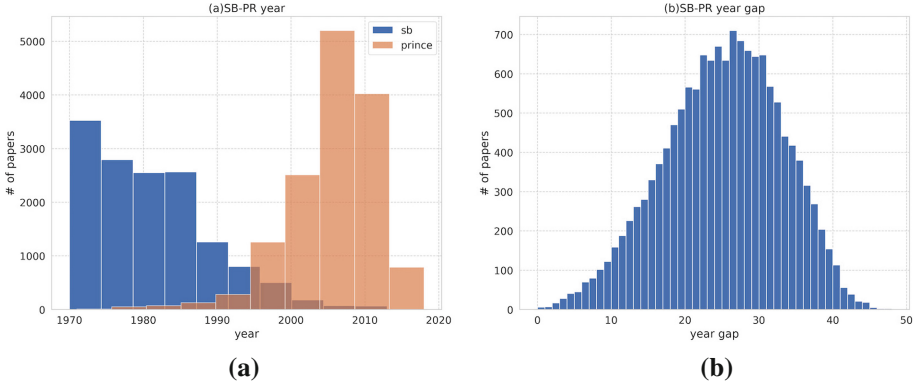


Fig. 1. (a) Annual distribution of SB and PR. (b) Gap year distribution of the SB paper and the PR paper.

We have defined the density of pairs D as follows:

$$A_{i,j}^y = \sum_y A_{y,i,j}. \quad (4)$$

$$D_{y,c_i,c_j} = \frac{A_{i,j}^y}{|c_i||c_j|} \quad (i \neq j) \quad (5)$$

In the above equation, $A_{y,i,j}$ indicates the number of papers in the cluster i that were published during the year y . Further, it also cites the papers in cluster j . Further, $|c_i||c_j|$ represents the possible edges between cluster i and cluster j , whereas $A_{i,j}^y$ showcases the actual edges between the two clusters until year y . When a PR published in the year y_p , and from the cluster c_p , cites the SB in cluster c_s , the density of this SB–PR paper is D_{y_p,c_p,c_s} . The density of the pair cannot be defined if the PR and SB are in the same cluster.

In this research, we have considered the first floor clustering of the entire citation network using the Leiden algorithm [16] as label for the papers. The purpose is to classify each paper into a unique category, as many papers exist in multiple disciplines these days.

Table 1 shows the example of each clusters. The top clusters include more than 8 million nodes, which are way too extensive to be considered under a single category. These may be covered under the basic concept of science. As we have specifically examined the cross-disciplinary SB–PR pairs in this study, we adopted the first floor clustering as a category to extract a more pointed cross section of the field. A more detailed analysis of the sub-clustering categories is necessary for future work.

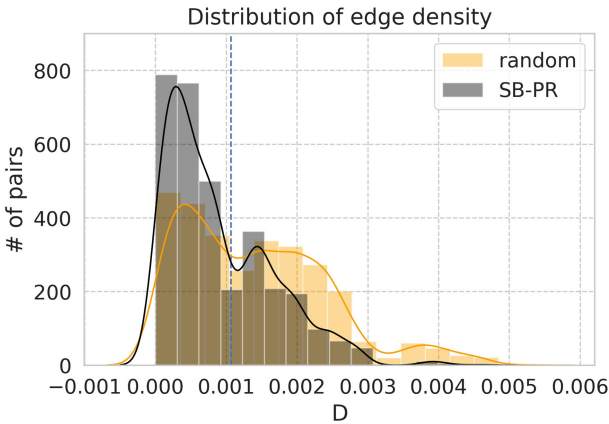
2.3 Density Distribution

Among the 14,317 pairs, only 1,857 pairs are a result of cross-disciplinary findings with a citation of an SB in another cluster. Therefore, most of the SB–PR

Table 1. Cluster size and detail of the top 10 largest clusters

Cluster	Size	Label	Frequent keywords
0	8,292,009	General	Education, China
1	5,564,222	Material Science	Microstructure, Mechanical properties
2	5,427,069	Informatics	Optimization, Simulation
3	3,866,952	Life-style related diseases	Obesity, Hypertension
4	3,703,869	Cancer	Brest Cancer, Apoptosis, Cancer
5	3,559,961	Biology	Taxonomy, New species
6	3,561,520	Intractable diseases	Alzheimer’s disease, Schizophrenia
7	2,944,362	Cell Biology	Appotosis, Asthma
8	2,947,362	Structural Chemistry	Crystal structure, synthesis

pairs are internal findings. Figure 2 presents the density distribution of cross-disciplinary SB–PR pairs. As compared to the random extraction from all cross-disciplinary citations, the distribution of SB–PR pairs is skewed to the left. This implies that SB–PR pairs include more rare collaborations than normal cross-disciplinary citations.

**Fig. 2.** Density distributions of SB and PR.

The distribution has two peaks. The first peak represents rare collaborations ($D < 1.07 \times 10^{-3}$). The most cross-disciplinary PR papers “explore” unusual categories of SB paper, thereby indicating that the PR broadens the possibilities of the field. The second peak represents common collaborations ($D \geq 1.07 \times 10^{-3}$). Even when similar papers are cited via common clusters, some PRs “rediscover” an important concept of SB papers. We have classified the bottom 66% of density under “exploring citations,” which are rare collaborations that transpired until that particular year. The other 33% are “rediscovering citations,” which re-evaluate the importance of common pairs of knowledge.

2.4 Rediscovering PRs and Exploring PRs

Publication of review papers frequently results in various scientific rediscoveries. Busy authors do not cite the original work; instead, they cite more recent derivative works and reviews [17]. The percentage of review papers for exploring PRs, overall PRs, and rediscovering PRs was 25%, 28%, and 35%, respectively, which increased at higher densities. Frequent citations between clusters led to the rediscovery of key findings.

Additionally, when we studied how PR papers cite SBs, we found out that discovering PRs are more likely to cite SBs in the Introduction and Results sections, whereas exploring PRs cite SBs in the Methodology section (Table 2). The introduction presents a brief description of the trajectory on which the research is based. It plays an important role in the early stage of research. Additionally, the Results section discusses core contributions toward the knowledge frontier. As a result, rediscovery of papers is presumed to extract research pairs that are linked strongly at the conceptual level. Citations in papers' Methodology section typically require an uncommon method to break the known challenges in the PR field. An SB category develops a way to solve other problems, which can be transferred to PR field problems. Moreover, among the top 100 PRs, 9 exploring PRs awaken multiple SBs, while all rediscovering PRs evoke only 1 SB. Exploring PRs have the potential to discover more than one SBs at a time.

Table 3 presents the highest and lowest examples of citation of two types of PR. Rediscovering PRs and SBs depict the field background and the comparison between the impact of the experimentally obtained results and results obtained from general studies. Exploring PRs are often used to conduct analyses that involve implementing methods that are not often used in a field. This paper has led to the popularization of this particular method of analysis in the field because this is the largest co-cited pair.

Table 2. Citation points of PR from SB for 100 articles each

Doctype	Citation point	Exploring PR	Rediscovering PR
Article, conference paper	Introduction	17	20
	Methodology	19	12
	Results, Discussion	6	19
	Others	9	3
Review		34	39
Book		14	5
Others		1	2
Total		100	100

2.5 Relation Type of SBs and Princes

Next, we identify whether the trend in SB-PR pairs varies by field. Figure 3 shows the specific rediscovering and exploring pairs that are more likely to occur

Table 3. Examples of exploring PR and rediscovering PR

D	Part	Citation sentence in PR
4.2×10^{-3} (rediscovering)	Introduction	Mitochondria are evolutionary endosymbionts derived from bacteria and contain DNA similar to bacterial DNA [19]
4.0×10^{-3} (rediscovering)	Result	This suggestion is surprising, because it is generally thought that chromatin structure does not play an important role in HSV gene transcription, largely because, unlike other viruses (e.g., SV40), newly replicated HSV genomes are not packaged into chromatin [20]
6.0×10^{-6} (exploring)	Methodology	Models with an initial percolating k-core cluster of quasi-crystalline short-range order showed shear localization at low strain rates; those without this order showed homogeneous deformation [21]
7.0×10^{-6} (exploring)	Methodology	LEfSe is implemented in Python and makes use of R statistical functions in the coin and MASS libraries through the rpy2 library and of the matplotlib library for graphical output [22]

between disciplines. Unlike exploring pairs, rediscovering PRs contribute largely to locally specific discipline SBs. For example, lifestyle-related diseases, cancer, cell biology, and molecular biology PRs tend to rediscover the past findings. These categories expand the specific knowledge range by leveraging references from closely related fields. In contrast, general, informatics, and materials engineering PRs are likely to use exploring citations. These clusters combine various types of knowledge through broader categories. It could be an intersection of scientific findings.

Instead of being explored, material science is more likely to explore various types of fields, indicating that the field applies key findings obtained from other fields. As far as informatics is concerned, it applies knowledge of the environment, materials engineering, and physical astronomy. Subsequently, biological categories, such as cancer and intractable diseases, make use of the findings. We can observe the circulation of knowledge across disciplines using citation rarity. This heatmap presents a foundation or relation type application of each pair of categories.

Table 4 presents the most frequent SB–PR pairs for each finding. The disciplines that become SBs and the ones that become PRs are relative matters. Thus, the flow of knowledge is not necessarily restricted to one direction (i.e., toward the basic and applied disciplines). However, some trends exist in scientific findings among the categories. Informatics may include key PRs that explore unknown knowledge from various fields, such as physics, materials engineer-

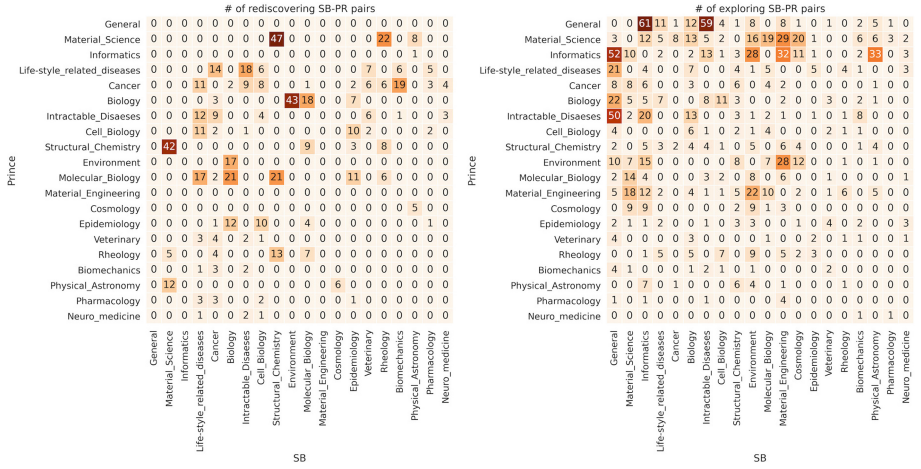


Fig. 3. Frequency of SB–PR pairs among the top 20 clusters.

ing, and environment related. Biology and chemistry, which are closely related, demonstrate rediscovery of the core concepts of the mutual findings.

Table 4. Frequent pairs of SB–PR in exploring and rediscovering collaborations

Exploring citation		Rediscovering citation	
PR	SB	PR	SB
General	Informatics	Materials Science	Rheology
General	Intractable Diseases	Materials Science	Structural Chemistry
Informatics	General	Biology	Environment
Informatics	Physics	Biology	Molecular Biology
Informatics	Materials Engineering	Structural Chemistry	Materials Science
Informatics	Environment	Molecular Biology	Biology
Intractable Diseases	General	Molecular Biology	Structural Chemistry
Materials Science	Materials Engineering	Molecular Biology	Lifestyle-related disease
Environment	Materials Engineering	Cancer	Biomechanics
Materials Engineering	Environment	Lifestyle-related disease	Intractable disease

2.6 Density vs. Citation

We hypothesize that, as an increasing number of exploration of citations occurs, the volatility of citation of PR papers increases because a rare combination unexpectedly produces revolutionary effects on research in the concerned field. However, the length of SB–PR pairs does not correlate with the citation of SBs ($R^2 = 0.00$) and PRs ($R^2 = 0.00$). We expected the citation gap, which separates

the successful papers from the failed papers, to be larger for exploring citations. However, the variance did not differ on the basis of whether the cited works were explored or rediscovered.

Furthermore, examination of key papers related to Nobel prize-winning findings selected by Mr. John Ioannidis [18] revealed that Nobel prize papers among the cross-disciplinary SB and PR papers are very few. We hypothesize that Nobel prize papers broaden the horizon of a category and they have an extremely strong impact beyond the representation of citations. Therefore, some of them may exist in SB–PR pairs. However, all SB–PR pairs include only four SBs and four PRs; cross-disciplinary pairs include only 1 SB. There was no correlation found between the impact of SB–PR papers and their density of citation. These results imply that surprising citations may not necessarily result in useful findings for the scientific community. With increasing attention being focused on the importance of cross-disciplinary research, the implications of the rarity of citations in the network are expected to be a major challenge in the future.

3 Conclusion

In this study, we have classified the types of SB–PR pairs across scientific disciplines in various fields. The relation of the pair is described on the basis of the citation rarity of the clusters that they are present in. The pairs have been broadly divided into two categories: major exploration citations and minor rediscovery citations. Rediscovering PRs contain more review articles than average. They refer to the SB in the Introduction and Results sections, which cite fundamentally important information about key findings. Meanwhile, the exploring PRs form an integral part of the Methodology section, which require an uncommon method to break the known challenges in the PR field. Furthermore, the materials science PRs, instead of being explored, are more likely to explore various types of fields, such as rheology or structural chemistry. This indicates that the field applies key findings obtained from other fields. However, biological subjects, such as cancer or cell biology, exhibit rediscovery of important papers through common clusters of SB–PR pairs.

This research contributes toward a better understanding of the delayed recognition across categories.

4 Data

We use bibliographic databases extracted from Scopus. These include 67 million papers and 1 billion citations from 27 fields covered from 1970 to 2018. The scientific fields are not fixed on the basis of time but rather expand and contract as and when they fuse and separate from other fields. Hence, we clustered the entire citation network into 1858 partitions using the Leiden algorithm [16] to identify the related category of each paper.

References

1. Garfield, E.: Premature discovery or delayed recognition - why? *Essays Inf. Sci.* **4**, 488–493 (1980)
2. Garfield, E.: Delayed recognition in scientific discovery: citation frequency analysis aids the search for case histories. *Curr. Contents* **23**, 3–9 (1989)
3. Garfield, E.: More delayed recognition. Part 2. From inhibin to scanning electron microscopy. *Essays Inf. Sci.* **13**, 68–74 (1990)
4. Campanario, J.M.: Rejecting and resisting Nobel class discoveries: accounts by Nobel Laureates. *Scientometrics* **81**(2), 549–565 (2009). <https://doi.org/10.1007/s11192-008-2141-5>
5. Fang, H.: An explanation of resisted discoveries based on construal-level theory. *Sci. Eng. Ethics* **21**(1), 41–50 (2015). <https://doi.org/10.1007/s11948-013-9512-x>
6. van Raan, A.F.J.: Sleeping beauties in science. *Scientometrics* **59**(3), 467–472 (2004). <https://doi.org/10.1023/B:SCIE.0000018543.82441.f1>
7. Mazloumian, A., Eom, Y.H., Helbing, D., Lozano, S., Fortunato, S.: How citation boosts promote scientific paradigm shifts and Nobel Prizes. *PLOS ONE* **6**(5) (2011). <https://doi.org/10.1371/journal.pone.0018975>
8. Ke, Q., Ferrara, E., Raduccgu, F., Flammini, A.: Defining and identifying sleeping beauties in science. *Proc. Nat. Acad. Sci. U.S.A.* **112**(24), 7426–7431 (2015). <https://doi.org/10.1073/pnas.1424329112>
9. van Raan AFJ.: Dormitory of physical and engineering sciences: sleeping beauties may be sleeping innovations. *PLOS ONE* **10**(10), e0139786 (2015). <https://doi.org/10.1371/journal.pone.0139786>
10. Du, J., Wu, Y.: A bibliometric framework for identifying “princes” who wake up the “sleeping beauty” in challenge-type scientific discoveries. *J. Data Inf. Sci.* **1**(1), 50–68 (2016). <https://doi.org/10.20309/jdis.201605>
11. Dey, R., Roy, A., Chakraborty, T., Chosh, S.: Sleeping beauties in computer science: characterization and early identification. *Scientometrics* **113**, 1645–1663 (2017). <https://doi.org/10.1007/s11192-017-2543-3>
12. Glänzel, W., Schlemmer, B., Thijs, B.: Better late than never? On the chance to become highly cited only beyond the standard bibliometric time horizon. *Scientometrics* **58**, 571–586 (2013). <https://doi.org/10.1023/b:scie.0000006881.30700.ea>
13. Costas, R., van Leeuwen, T.N., van Raan, A.F.J.: Is scientific literature subject to a ‘Sell-By-Date’? A general methodology to analyze the ‘durability’ of scientific documents. *J. Am. Soc. Inf. Sci. Technol.* **61**, 329–339 (2010). <https://doi.org/10.1002/asi.21244>
14. Li, J.: Citation curves of “all-elements-sleeping-beauties”: “flash in the pan” first and then “delayed recognition”. *Scientometrics* **100**(2), 595–601 (2013). <https://doi.org/10.1007/s11192-013-1217-z>
15. Ioannidis, J.P.A., Baas, J., Klavans, R., Boyack, K.W.: A standardized citation metrics author database annotated for scientific field. *PLOS Biol.* **17**(8) (2019). <https://doi.org/10.1371/journal.pbio.3000384>
16. Traag, V.A., Waltman, L., van Eck, N.J.: From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**, 5233 (2019). <https://doi.org/10.1371/journal.pbio.3000384>
17. Marks, M.S., Marsh, M.C., Schroer, T.A., Stevens, T.H.: An alarming trend within the biological/biomedical research literature towards the citation of review articles rather than the primary research papers. *Traffic* **14**(1), 1 (2013). <https://doi.org/10.1111/tra.12023>

18. Ioannidis, J.P.A., Cristea, I.A., Boyack, K.W.: Work honored by Nobel prizes clusters heavily in a few scientific fields. *PLOS ONE* **15**(7), e0234612 (2020). <https://doi.org/10.1371/journal.pone.0234612>
19. Oka, T., Hikoso, S., Yamaguchi, O., Taneike, M., Takeda, T., Tamai, T., Akira, S.: Mitochondrial DNA that escapes from autophagy causes inflammation and heart failure. *Nature* **485**(7397), 251–255 (2012). <https://doi.org/10.1038/nature10992>
20. Wysocka, J., Myers, M.P., Laherty, C.D., Eisenman, R.N., Herr, W.: Human Sin3 deacetylase and trithorax-related Set1/Ash2 histone H3–K4 methyltransferase are tethered together selectively by the cell-proliferation factor HCF-1. *Genes Dev.* **17**(7), 896–911 (2003). <https://doi.org/10.1101/gad.252103>
21. Schuh, C.A., Hufnagel, T.C., Ramamurty, U.: Mechanical behavior of amorphous alloys. *Acta Mater.* **55**(12), 4067–4109 (2007). <https://doi.org/10.1016/j.actamat.2007.01.052>
22. Segata, N., Izard, J., Waldron, L., Gevers, D., Miropolsky, L., Garrett, W.S., Huttenhower, C.: Metagenomic biomarker discovery and explanation. *Genome Biol.* **12**(6), 1–18 (2011). <https://doi.org/10.1186/gb-2011-12-6-r60>



Finding High-Degree Vertices with Inclusive Random Sampling

Yitzchak Novick^{1,2}  and Amotz BarNoy³

¹ CUNY Graduate Center, New York, NY 10016, USA
ynovick@gradcenter.cuny.edu

² Touro College and University System, New York, NY 10018, USA

³ Brooklyn College, Brooklyn, NY 11210, USA
amotz@sci.brooklyn.cuny.edu

Abstract. The friendship paradox (FP) is the famous phenomenon that one's friends typically have more friends than they themselves do. The FP has inspired novel approaches for sampling vertices at random from a network when the goal of the sampling is to find vertices of higher degree. The most famous of these methods involves selecting a vertex at random and then selecting one of its neighbors at random. Another possible method would be to select a random edge from the network and then select one of its endpoints at random, again predicated on the fact that high degree vertices will be overrepresented in the collection of edge endpoints. In this paper we propose a simple tweak to these methods where we consider the degrees of the two vertices involved in the selection process and choose the one with higher degree. We explore the different sampling methods theoretically and establish interesting asymptotic bounds on their performances as a way of highlighting their respective strengths. We also apply the methods experimentally to both synthetic graphs and real-world networks to determine the improvement inclusive sampling offers over exclusive sampling, which version of inclusive sampling is stronger, and what graph characteristics affect these results.

Keywords: Inclusive random sampling · Random neighbor · Friendship paradox

1 Introduction

1.1 Random Neighbor

It is often of interest to locate vertices within a network that are of relatively high degree (see for example [1, 5, 7]). But in most networks, there is no obvious way to do this efficiently, either because total network knowledge does not exist, or because it is unavailable due to the network's size and dynamic nature. Consider the most obvious random sampling method of naively sampling a random vertex, which we will abbreviate as RV . The expected degree of a vertex obtained by RV is simply the mean degree of the graph, or $RV = \mu^1$.

¹ We will interchangeably use method abbreviations to refer to the method and the expected degree of a vertex it returns.

Cohen et al. [6] offer a novel approach that leverages the well know “friendship paradox” (FP) [9] to give a higher expected degree vertex than RV . The friendship paradox is the phenomenon that a person’s friends will, on average, have more friends than the person themselves does. The method of selecting a random neighbor (RN) is performed by selecting a random vertex as in RV , but then taking the collection of its neighbors and selecting one of these neighbors instead of the originally selected vertex. Because the “friends” have a higher degree on average, the expected degree of RN should be greater than or equal to the expected degree of RV , or $RN \geq RV$. The superiority of RN has been demonstrated by [6] and [12], and a formal proof appears in [10] where it is also attributed to a comment in an online article [14].

1.2 Random Edge

In a paper that is still in progress [10], Kumar et al. differentiate between two mean degrees of a graph that are both inspired by the friendship paradox. The first is the ‘local mean’, which is calculated by taking every vertex individually, finding the mean degree of its neighbors, and then taking the mean value of this mean over all vertices. Clearly the local mean is perfectly analogous to RN .

The second mean they discuss is the ‘global mean’. The global mean is the mean degree of the collection of all neighbors in the graph, which is compiled by taking the collection of neighbors of every individual vertex and combining these collections into a single collection. Notably, vertices with degree 2 or higher will appear in this collection multiple times. In fact, they will appear exactly as many times as their degree.

The authors offer a clever sampling technique whose expectation is the global mean of the graph, by examining all neighbors of a randomly selected vertex and selecting each with some fixed probability p . By considering all neighbors of a selected vertex, the probability of a vertex being considered is directly proportional to its degree, and all vertices that are considered have equal probability (p) of being selected. Therefore, the expected degree of this sampling method is exactly the global mean.

We note here that the global mean can also be achieved by a different sampling method. A single edge from the collection of edges in the graph can be selected at random, then one of its two endpoints can be selected at random. Once again, a vertex’s likelihood of being selected is proportional to the number of edges it touches, in other words its degree. We will call this method ‘random edge’ (RE). The probabilistic method of [10] has strong practical appeal because the edges of a graph are rarely stored as a separate collection. Typically, an edge could only be found by selecting a vertex and then identifying its neighbors. (Perhaps a road network would be a real-world exception.) However, as this paper is an academic exploration of the expected values rather than a discussion of implementation or practicality, we prefer to frame our results in the context of RE rather than the probabilistic method, while recognizing that all of the authors’ findings for the global mean apply equally to RE and vice versa.

Based on the authors’ results for the global mean, $RE \geq RV$ can be proven directly from the FP. It is interesting that RE is actually the purer manifestation of the FP as a sampling method; $RN \geq RV$ is not in fact directly implied by the FP. It is also interesting to note where each method reduces to RV . $RE = RV$ only in a regular graph, whereas

$RN = RV$ in a completely assortative graph (as defined in [13]), even if the degrees of the vertices are not all identical.

1.3 Inclusive Sampling

The primary contribution of this paper is to propose a tweak to both RN and RE . Instead of blindly taking the randomly selected neighbor in the case of RN , or selecting the edge-endpoint at random in the case of RE , we would compare the respective degrees of the two vertices, the original vertex and the neighbor in RN , and the two edge-endpoints in RE , and select the one with higher degree. It should be noted that this is not a purely hypothetical suggestion. Even in networks where the lack of total knowledge prevents one from selecting high-degree vertices directly, it is still typically possible to know the degree of any selected vertex. In most cases this value would be stored separately, the collection of neighbors would not even need to be enumerated. Even in an offline human network it is not hard to conceive of a scenario where two selected individuals would consent to having their phones scanned by software that would give some acceptable estimate of their popularity based on their contacts, emails, social media activity, etc. We will call these methods ‘inclusive random neighbor’, or IRN , and ‘inclusive random edge’, or IRE .

2 Sampling Method Comparisons

2.1 Calculating the Expectations

We begin this study with a direct comparison between RN and RE . Kumar et al. [10] demonstrated that it is possible for RN or RE to have the higher expected degree in a graph. We will look at the ratio $RN : RE$ as well as the inverse ratio, $RE : RN$, and show that both ratios can grow without bound.

In performing this study, we opt to ignore any vertices with degree 0. It is not clear what RN would even do should a 0-degree vertex be selected. In RN would select the 0-degree vertex there would be an obvious advantage to RE because it does not even consider these vertices. But regardless of how RN would deal with 0-degree vertices, we consider a study of RN vs. RE more interesting if we only include the subgraph that contains edges, so that RN and RE are sampling from the same set of vertices. We will therefore simply ignore any vertices without neighbors.

In the following equations, we will use n to denote the number of vertices in the graph, and V is the collection of vertices itself. Similarly, m will denote the number of edges in the collection of edges E . We will use Λ_v to denote the collection of the neighbors of vertex v . An edge between u and v will be denoted $e(u, v)$. We will consider D the degree sequence of the graph and use d_v to denote the degree of vertex v .

RE can be defined as:

$$RE = \frac{1}{m} \sum_{e(u,v) \in E} \frac{d_u + d_v}{2} \quad (1)$$

RN can be defined as:

$$RN = \frac{1}{n} \sum_{v \in V} \frac{1}{d_v} \sum_{u \text{ in } \Lambda_v} d_u \tag{2}$$

It is worth noting that the contribution of every edge $e(u, v)$ to the outer summation is $\frac{d_u}{d_v} + \frac{d_v}{d_u}$ and therefore RN can also be expressed as a summation over E .

$$RN = \frac{1}{n} \sum_{e(u,v) \in E} \frac{d_u}{d_v} + \frac{d_v}{d_u} \tag{3}$$

The ratios of RN to RE can therefore be written as:

$$\frac{RN}{RE} = \frac{2m \sum_{e(u,v) \in E} \frac{d_u}{d_v} + \frac{d_v}{d_u}}{\sum_{e(u,v) \in E} d_u + d_v} \tag{4}$$

And its inverse:

$$\frac{RE}{RN} = \frac{n \sum_{e(u,v) \in E} d_u + d_v}{2m \sum_{e(u,v) \in E} \frac{d_u}{d_v} + \frac{d_v}{d_u}} \tag{5}$$

Corollary 1. $\frac{RN}{RE} \leq \frac{2m}{n}$.

Proof. Every edge contributes a value in the form of $\frac{a}{b} + \frac{b}{a}$ to the numerator of (4), and a value in the form of $a + b$ to the denominator.

$$\frac{a}{b} + \frac{b}{a} = \frac{a^2 + b^2}{ab} \leq a + b = \frac{a^2b + b^2a}{ab}$$

Corollary 2. $\frac{RN}{RE} < \frac{2m}{n}$ in all graphs with a single vertex v with $d_v > 1$.

Proof. There exists at least one edge (u, v) with $d_u > 1$. If $a > 1$ and $b \geq 1$ then

$$a^2 + b^2 < a^2b + b^2a$$

If we assume without loss of generality that $d_u \geq d_v$, we can define IRN as:

$$IRN = \frac{1}{n} \sum_{e(u,v) \in E} \frac{d_u}{d_v} + 1 \tag{6}$$

And IRE as:

$$IRE = \frac{1}{m} \sum_{e(u,v) \in E} d_u \tag{7}$$

Corollary 3. $\frac{IRE}{RE} < 2$.

Proof. For every edge $e(u, v)$ with $d_u \geq d_v$, because $d_v \geq 1$, $d_u < 2\frac{d_u+d_v}{2} = d_u + d_v$.

2.2 Strengths of RN and RE

In order to maximize one of the ratios, we construct a graph that accentuates the strength of the sampling method in the numerator. We consider a graph that is comprised of two disconnected subgraphs with h and k vertices respectively. The first subgraph will have the majority of the graph’s edges, so that RE will select a vertex from this subgraph with high probability. Similarly, the second subgraph has the majority of the vertices, so that RN will select a vertex from this subgraph with high probability. In both cases, we make the first subgraph a clique, saturating it with edges. If we want RE to be the superior method, we lower the degrees of the k vertices in the second subgraph by arranging them as a collection of edges connecting two 1-degree vertices. This is actually a generalization of a figure in [10] that demonstrates that it is possible for $RE > RN$. If we want RN to be stronger we arrange the k vertices of the second subgraph into a star. If we select a vertex from the second subgraph, with probability $(k - 1)/k$, we will select a leaf whose only neighbor is degree $k - 1$ (Fig. 1).

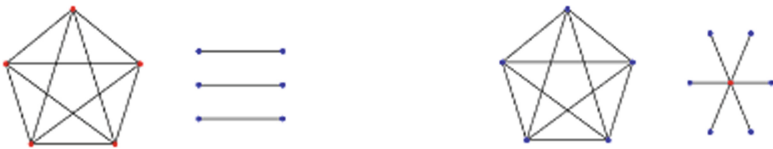


Fig. 1. Two constructions that illustrate how either RN or RE could be made the stronger sampling method in a graph.

2.3 RE/RN and RN/RE Are Both Unbounded

RE/RN is Unbounded. It is possible for RE/RN to be arbitrarily large. Consider a graph with a clique of h vertices and $k/2$ edges linking k 1-degree vertices:

$$\frac{RE}{RN} = \frac{(h(h - 1)^2 + k)(h + k)}{(h(h - 1) + k)^2} \tag{8}$$

This expression grows without bound as h/k increases.

This expression can also be used to give a possible lower bound on RE/RN as a function of n . In other words, it answers the question: ‘how large a graph is required to achieve a desired ratio?’. If we set $k = h(h - 1)$ then $n = h + h(h - 1) = h^2$. Rewriting (8) in terms of h gives:

$$\frac{(h(h - 1)^2 + h(h - 1))(h + h(h - 1))}{(h(h - 1) + h(h - 1))^2} = \frac{h^2}{4(h - 1)} = \Omega\left(n^{\frac{1}{2}}\right) \tag{9}$$

Notice that, by symmetry, IRE and IRN reduce to RE and RN respectively in this construction. Therefore, these proofs suffice to prove the same results for IRE/IRN .

RN/RE is Unbounded. We can follow a similar process as the one above to prove that RN/RE is unbounded in the clique and star graph by setting $h = x^2$ and $k = x^3 - x^2$. This will give an expression that grows without bound as k/h increases. It can further be used to give a bound on the ratio as a function of n as $\Omega(n^{1/3})$. Here too it is possible to prove that the results apply to IRN/IRE as well. We omit the full steps here in the interest of brevity.

2.4 RE and RN in Trees

We will now examine our sampling methods in the specific case of trees.

RN/RE is Bounded by 2. Our first observation is that RN must be less than $2RE$. This follows from Corollary 1 because $2m/n$ is fixed at $2((n - 1)/n)$. It would also seem that a star maximizes RN/RE for all trees of a fixed size n . For a star:

$$\frac{RN}{RE} = \frac{2(n - 2)^2 + 2}{n^2}$$

This expression has the same bound, 2, as n increases.

RE/RN is Unbounded in Trees. However, RE/RN is still unbounded, even in trees. Consider a tree with a root that has h children, and each child has an additional $k - 1$ children, all of which are leaf nodes (Fig. 2).



Fig. 2. A graph tree with h internal nodes and $h(k - 1)$ leaf nodes.

If we fix k and increase h so that $h \gg k$, RE approaches $h/2k$ and RN approaches h/k^2 . This ratio converges on $k/2$ and can therefore be arbitrarily large.

Using a technique similar to the one we used to establish previous bounds as a function of n , we can prove that, if $k^3 < h$, the bound in this construction is $\Omega(n^{1/4})$.

3 Sampling Methods in Synthetic and Real-World Graphs

We calculated values for RN , RE , IRN , and IRE in both synthetic and real-world graphs. For synthetic graphs, we looked at both Erdős Rényi (ER) [8] and Barabási Albert (BA) [2] models, using different parameters and taking the mean results of 30 randomly generated graphs for each parameter set. For real-world graphs, we examined networks from the famous Koblenz Network Collection [11].

3.1 Synthetic Graphs

In both ER and BA graphs a very interesting trend emerges. In both types, as would be expected, $RN > RV$ and $RE > RV$, because the graph almost certainly will contain at least one edge connecting two vertices of different degree. The gains for both of the methods over RV was modest in ER graphs but significant in BA graphs. In ER graphs, RN was always minimally better than RE . In BA graphs this was almost always true as well, but when the edge count was very high $RE > RN$, which is seemingly consistent with our analysis from Sect. 2 because of the emergence of more substructures with high concentration of edges similar to the cliques in those examples. We speculate that the fact that RN was so much stronger in BA graphs is linked to assortativity. As we will discuss in Sect. 4, disassortativity in a graph increases RN . Although both ER and BA graphs tend to be non-assortative [13], it has been demonstrated [3, 4] that in BA graphs this is only an aggregate result of the fact that many low-degree vertices are very assortative while the high-degree vertices are very disassortative. The disassortative part of the graph is enough to improve RN in BA graphs compared to ER graphs where the graph is more homogeneously nonassortative.

The inclusive sampling methods revealed perhaps the most interesting result. Obviously, the assumptions $IRN > RN$ and $IRE > RE$ held. But, despite the fact that it was almost always true that $RN > RE$, it was always true that $IRE > IRN$. We summarize the findings for graphs with $n = 6000$ in Table 1 below.

Table 1. Sampling method results for ER/BA graphs, $n = 6000$

	Erdős Rényi Graphs, $n = 6000$				Barabási Albert Graphs, $n = 6000$			
RV	RN	RE	IRN	IRE	RN	RE	IRN	IRE
6	6.9952	6.9946	7.9227	8.361	19.54	17.68	21.34	29.63
10	10.9883	10.9882	12.3023	12.755	27.87	26.18	30.7	42.76
16	16.973	16.9714	18.7509	19.2119	38.9	37.43	43.24	59.46
30	30.922	30.9212	33.525	33.9967	63.89	62.75	71.64	96.63
60	60.6866	60.6864	64.5381	65.0121	113.3	112.55	128.18	167.78
129	129.5657	129.565	135.4022	135.8766	216.32	216.42	246.99	310.69

These findings again reflect on the respective natures of the sampling methods. RE is a pure manifestation of the FP, it relies entirely on the fact that high-degree vertices are overrepresented in a collection of edge endpoints. But, between the two edge endpoints of a given edge, there is no favoring one vertex over the other. On the other hand, RV seems to be implicitly assuming that the jump from a vertex to a neighbor will lead to an increase in degree. In most natural graphs, the low degree vertices will outnumber the high-degree vertices so RN is actually a type of correction to RV , improving the outcome by exchanging the random vertex for its neighbor. Therefore, in RN , the gain of inclusive sampling is less significant. It only applies in the less common case where a high-degree vertex was selected in the first step of the process. Whereas in RE the inclusive sampling is more significant because the mean degree of the two edge endpoints is always less than or equal to the max degree.

3.2 Real-World Networks

We examined 1072 networks from the Koblenz Network Collection to see the effects of the four sampling methods. We found that $RN > RE$ in 93% of the networks, yet $IRE > IRN$ in 43%. The average gain of IRN versus RN was 102.3%, while the average gain of IRE versus RE was a staggering 186%. This is especially significant in light of the bound of 2 in Corollary 3.

We also calculated these results for the different network categories of the collection. The results are summarized in Table 2. $RN > RE$ is true in the majority of networks in all but three categories, and the mean percent over all categories where this is true is 72.8%. $IRE > IRN$ is true for a majority of networks in all but three categories (note that these are not the same three categories where $RE > RN$), and the mean percent over all categories where this is true is 82.2%. The modest gains of IRN over RN are roughly consistent over all categories, while the gain of IRE over RE ranges from 1.13 to 1.98.

Table 2. Method comparisons in real-world networks by category

Category	Pct $RN > RE$	Pct $IRN > IRE$	IRN/RN	IRE/RE
Affiliation	100%	17%	1.05	1.68
Animal	75%	0%	1.09	1.13
Authorship	99%	67%	1.01	1.94
Citation	50%	0%	1.08	1.58
Cocitation	0%	0%	1.1	1.47
Communication	83%	25%	1.04	1.7
Computer	64%	0%	1.07	1.60
Feature	83%	50%	1.02	1.88
Human contact	86%	14%	1.12	1.31
Human social	55%	0%	1.12	1.21
Hyperlink	71%	14%	1.02	1.84
Infrastructure	48%	0%	1.1	1.2
Interaction	81%	62%	1.04	1.71
Lexical	67%	33%	1.08	1.66
Metabolic	75%	0%	1.07	1.59
Misc	67%	0%	1.08	1.55
Neural	100%	0%	1.11	1.45
Online contact	75%	13%	1.03	1.69
Rating	100%	57%	1.02	1.87
Social	71%	31%	1.03	1.76
Software	100%	67%	1.003	1.98
Text	83%	0%	1.04	1.58
Trophic	100%	0%	1.14	1.33

4 Inclusive Sampling Methods and Degree Homophily

For *RN* and *RE*, [10] already established that degree homophily affects the results. We perform a similar analysis on the inclusive sampling methods. The authors in [10] introduce a new measure of degree homophily, inversity, and show that its sign perfectly predicts which of *RN* or *RE* will be higher. But, as we are only looking for trends and not precise predictions, we will use the better known assortativity [13] for this analysis. While the authors demonstrated that assortativity and inversity do not capture precisely the same information, they are very closely correlated. The authors find that less degree homophily causes *RN* to strengthen relative to *RE*.

We generated two sets of graphs, one ER and one BA, using the same parameters for all graphs in each set. We then rewired the graphs, following a process that is detailed in [15, 16] to achieve higher and lower assortativity values while retaining the degree sequences, and measured the values of the sampling methods with each rewiring. It is worth noting that, because assortativity and all four sampling methods are functions of summations over the edge collection, all values can be adjusted in $O(1)$ time with each rewiring rather than being recalculated from scratch. A typical result for both ER and BA graphs is displayed in Fig. 3 below.

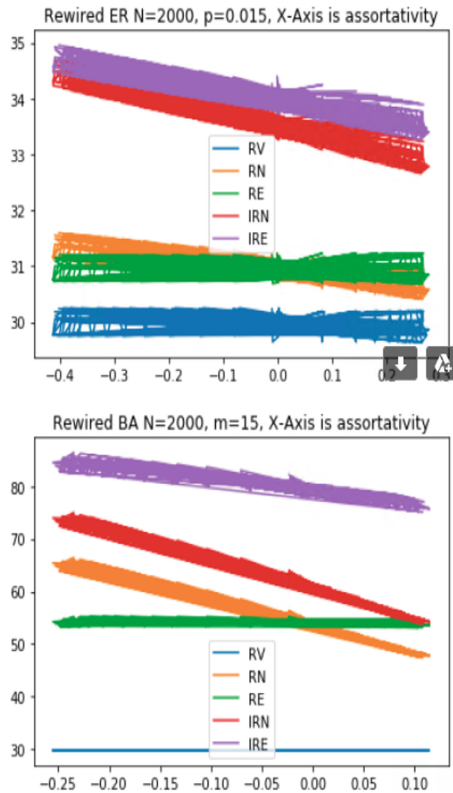


Fig. 3. Scatter plots of sampling method results by assortativity

RE is purely a function of the degree sequence, so rewiring cannot affect it, and this is clear in the plots (any noise presumably reflects an inability of some degree sequences to achieve certain assortativity values). RN becomes weaker as assortativity increases, and the fact that these two lines meet around zero assortativity is consistent with the fact that the sign of inversity, which is very close to assortativity, indicates the strength of RN relative to RE .

The plots again indicate a strength in IRE over IRN , and while both weaken as assortativity increases, it appears that IRN decreases at a slightly faster rate.

5 Summary and Future Research Directions

In this paper we explored two sampling methods that leverage the phenomenon of the FP to perform random sampling with a bias towards higher degree vertices. We proved that either method can be infinitely better than the other and gave possible lower bounds on the number of vertices that would be required in order to achieve a desired ratio.

We introduced “inclusive” versions of each of these methods and showed a surprising result that IRE is often greater than IRN , even in graphs where RN outperforms RE . While we explored these methods mostly as an academic study, we noted that inclusivity itself is not a contrived concept considering the fact that, once a vertex in a network has been selected, its degree is typically available. We believe this study can have strong practical applications in situations where high-degree sampling is desired. While edges are not typically stored as collections from which to draw samples, we noted the existence of a probabilistic sampling method that takes a similar approach to RN but achieves the results of RE . Furthermore, perhaps in some situations where high-degree sampling is often required and the nature of the graph makes RE a significantly stronger options, it would actually be worthwhile to store edges in order to enable RE sampling.

While we have noted the strong connection with degree-homophily, we believe there are other graph characteristics, for example the power-law exponent, that contribute to the results of the different methods and we hope to explore some of these in future research. We also note that we have evaluated the methods based solely on their results. We are currently studying the methods in light of not only results, but also the computational complexity and other costs that could be associated with them in order to give a far more robust analysis of their respective values as high-degree sampling methods.

References

1. Albert, R., Jeong, H., Barabási, A.L.: Error and attack tolerance of complex networks. *Let. Nat.* **406**, 378–382 (2000)
2. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* **286**, 509–512 (1999)
3. Babak, F., Rabbat, M. G.: Degree Correlation in Scale-Free Graphs, [arXiv:1308.5169](https://arxiv.org/abs/1308.5169) (2013)
4. Bertotti, M.L., Modanese, G.: The Bass Diffusion Model on Finite Barabasi-Albert Networks, *Phys. Soc.* [arXiv:1806.05959](https://arxiv.org/abs/1806.05959) (2018)
5. Cohen, R., Erez, K., Ben-Avraham, D., Havlin, S.: Breakdown of the internet under intentional attack. *Phys. Rev. Lett.* **86**(16), 3682–3685 (2001)

6. Cohen, R., Havlin, S., Ben-Avraham, D.: Efficient immunization strategies for computer networks and populations. *Phys. Rev. Lett.* **91**(24), 247901 (2003)
7. Christakis, N.A., Fowler, J.H.: Social network sensors for early detection of contagious outbreaks. *PLoS ONE* **5**(9), e12948 (2010)
8. Erdős, P., Rényi, A.: *Publicationes Mathematicae* **6**(290) (1959)
9. Feld, S.: Why your friends have more friends than you do. *Am. J. Soc.* **96**(6), 1464–1477 (1991)
10. Kumar, V., Krackhardt, D., Feld, S.: Network Interventions Based on Inversity: Leveraging the Friendship Paradox in Unknown Network Structures (2018). <https://vineetkumars.github.io/Papers/NetworkInversity.pdf>
11. Kunegis, J.: KONECT, The Koblenz Network Collection (2013). <http://konect.cc/>
12. Momeni, N., Rabbat, M.G.: Effectiveness of Alter Sampling in Social Networks, <https://arxiv.org/abs/1812.03096v2> (2018)
13. Newman, M.E.J.: Assortative mixing in networks. *Phys. Rev. Lett.* **89**(20), 208701 (2002)
14. Strogatz, S.: Friends You Can Count On, NY Times (2012). Accessed 17 Sep 2012. <https://opinionator.blogs.nytimes.com/2012/09/17/friends-you-can-count-on/>
15. Van Mieghem, P., Wang, H., Ge, X., Tang, S., Kuipers, F.A.: Influence of assortativity and degree-preserving rewiring on the spectra of networks. *Eur. Phys. J. B* **76**, 643–652 (2010)
16. Xulvi-Brunet, R., Sokolov, I.M.: Reshuffling scale-free networks: from random to assortative. *Phys. Rev.* **70**, 066102 (2004)



Concept-Centered Comparison of Semantic Networks

Darkhan Medeuov^{1,3(✉)}, Camille Roth², Ksenia Puzyreva¹, and Nikita Basov¹

¹ Centre for German and European Studies, St. Petersburg University,
Universitetskaya nab. 7/9, St. Petersburg 199034, Russia

darkhan.medeuov@nu.edu.kz

² Centre Marc Bloch, Humboldt Universität Berlin, Berlin, Germany

³ Nazarbayev University, Nur-Sultan, Kazakhstan

Abstract. This article proposes an approach to compare semantic networks using *concept-centered* sub-networks. We illustrate the approach on written and interview texts from an ethnographic study of flood management practice in England.

Keywords: Semantic networks · Text analysis · Flood management

1 Introduction

Research in cultural sociology argues that semantic networks –with vertices being concepts and links being concept co-occurrences within a given time-window– are capable of revealing structures of knowledge underpinning production of texts (Carley 1986; Carley and Newell 1994; Lee and Marin 2015) and thus help to explore these structures through the lens of network analysis (Abbott et al. 2015; Roth and Cointet 2010). Currently, an issue of particular interest is how different knowledge systems (for instance institutional-field and local knowledge, see Basov et al. 2019) interplay with each other. This paper proposes an approach to examine this interplay at the meso-level of particular concepts which gain different meanings across different knowledge systems.

We draw on a new textual dataset on professional and local flood management knowledge collected during one of this paper’s authors ethnographic study in England. Flood management in England as a knowledge domain provides an apt example because, until recently, it exclusively relied on institutionalized professional knowledge. In recent decades, however, flood management has become more sensitive to ‘local knowledges’ and started seeking ways to engage local actors (McEwen and Jones 2012; Nye et al. 2011; Wehn et al. 2015). Becoming stakeholders in flood risk management local, communities/activists are expected to adhere to professional knowledge and language. They, however, rarely take professional knowledge at ‘face value’, but rather creatively reuse it to fit the local context (Nye et al. 2011; Wehn et al. 2015). Our data comes from one such flood-prone area in England where flood management agencies and local

community groups collaborate to manage flood risks. We examine what professional concepts are used by local actors. We represent professional knowledge as a semantic network and then examine concept-centered sub-networks shared by both the professional and local semantic networks.

The paper is organized as follows. In the next section, we introduce the idea of concept-centered sub-networks and lay out reasons why researchers might better understand semantic networks focusing on their concept-centered components. We describe a two-step analytical approach first showing how to find potentially interesting concept-centered sub-networks and then how to highlight their similarities and differences using a customized version of the Fruchterman-Reingold force-directed layout (Fruchterman and Reingold 1991). We then describe our data and illustrate the approach. We conclude by discussing the main results and outlining future work prospects.

2 Semantic Networks and Concept-Centered Networks

A semantic network, as any other network, is defined as a couple of sets $G = \{V, E\}$. V lists all the nodes in a network, and $E \in V \times V$ lists all the connected pairs of nodes. A network can also be defined by an adjacency matrix A of dimension $|V| \times |V|$ with an entry a_{ij} indicating a link between nodes i and j . In the most simple case the matrix A is binary and symmetric: $a_{ij} \in \{0, 1\}$, $a_{ij} = a_{ji}$. Semantic networks, however, can include co-occurrence counts and the entries in their adjacency matrices are not binary but positive integers, $a_{ij} \in \mathbb{Z}^+$. For this reason, researchers often binarize co-occurrence matrices using some threshold value, τ , $(a_{ij} \geq \tau \rightarrow a_{ij} = 1) \wedge (a_{ij} < \tau \rightarrow a_{ij} = 0)$ (Cantwell et al. 2020; Dianati 2016). We follow this thresholding approach in this paper.

Let us consider two binary semantic networks, A_1 and A_2 . For clarity reasons, we will refer to these networks as if they belong to two actors, actors a_1 and a_2 . There exist many ways to compare networks, however, we argue that semantic networks specifically may be better understood by comparing different substructures they consist of (e.g. network motifs) (Choobdar et al. 2012; Milo et al. 2002; Pržulj 2007). We argue that comparison of knowledge systems can be narrowed down to the use of particular concepts, which at the level of semantic networks corresponds to concept-centered sub-networks. For example, knowing that concept *flood* is linked to different concepts in professional and local semantic networks¹ may hint that *flood* has different meanings in professional and local knowledge.

Looking across all concepts shared by two networks, we could map concepts linked with the same other concepts (alters). Then, we could expose these similarities using a specific layout. Visualization is an important tool for exploratory analysis as it assists in focusing on linkage patterns and generating hypotheses as to what causes concepts to be interlinked in particular ways.

¹ That is, the concept tends to co-occur with different other concepts in professional and local texts.

To summarize, we propose an approach to compare concept-centered networks by looking at the similarity of their vertex and edge sets.

We define a concept-centered network as follows. For a given actor a and given concept c the concept-centered network $C_{ac} = \{V_{ac}, E_{ac}\}$ is an induced network whose vertex set V_{ac} consists of the given concept (ego) and its adjacent concepts (alters) and whose edge set E_{ac} of all edges in E between pairs of vertices in V_{ac} .

We measure similarity using vertex and link overlap indices. In general, an overlap index shows how many shared elements two sets have relative to the cardinality of the smaller set, we denote such index as ω . We compute the vertex overlap index for two concept-centered networks excluding the ego concept, hence we denote corresponding vertex sets V_{ac}^* . When computing the link overlap index we also exclude ego-incident links, focusing only on alter-alter links, which we denote E_{ac}^* .

We choose to exclude ego and ego-incident links because keeping them induces a lower bound. While this lower bound is somewhat negligible for the vertex overlap, it can be substantial for the link overlap. We use overlap indices instead of the more common Jaccard indices because we assume that it is local actors that draw upon professional knowledge. In other words, we are interested in what part of concepts and linkages local actors draw on professional knowledge relative to their personal perspective².

$$\omega(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)} \quad (1)$$

$$\text{Vertex Overlap (concept } c, \text{ actors } a, a') = \omega(V_{ac}^*, V_{a'c}^*) \quad (2)$$

$$\text{Link Overlap (concept } c, \text{ actors } a, a') = \omega(E_{ac}^*, E_{a'c}^*) \quad (3)$$

The way we define vertex and link overlap indices assumes that the central concept c has at least one alter. Otherwise, there would be division by 0. We assume this because we are not practically interested in empty/trivial concept-centered semantic networks and only focus on the most central concepts which are necessarily non-trivial.

Along with the analytical approach we also propose a visualization approach to highlight similarities and discrepancies between two concept-centered networks. We first combine concept-centered sub-networks into a “union” network by gathering all vertices and links. The union graph helps to position nodes (e.g., in Fig. 1) and keep these positions thereafter to visualize each network separately. We assign weights to links with those shared by both actors having larger weight. The weights serve functional roles: we want shared links to weight more than non-shared ones because this way a force-directed layout makes vertices which are incident to shared links more attracted to the ego-concept and themselves. Using weights not only puts shared concepts closer to each other

² Which is always smaller than that of professionals, mostly due to different corpus sizes.

but also spatially separates actor-unique concepts that are connected to shared concepts from those that are not.

Figure 1 illustrates this idea. Suppose we have a pair of actors and their respective concept-centered networks. The ego-concept in both networks is $c1$. The first actor links the concept $c1$ to concepts $c2$, $c3$, $c4$, and $c5$. In the second network $c1$ is not linked to $c5$ but to $c6$. Besides, there is the difference in the alter-alter linkage: the first actor links $c2$ and $c5$, while the second actor links $c2$ and $c3$. This results in the vertex Jaccard similarity between these concept-centered networks being 0.75 (3 shared vertices out of 4 vertices). At the same time, the link overlap similarity between them is 0.5 (1 shared alter-alter links out of 2).

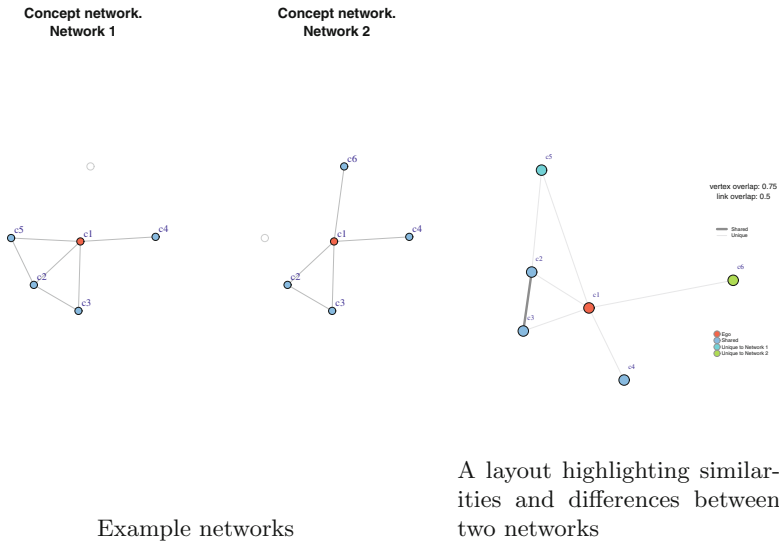


Fig. 1. Actors share 3 links between the concept $c1$ and other concepts ($c2$, $c3$, and $c4$) with the minimum number of unique concepts for two actors being 4. This makes for vertex overlap similarity of 0.75. The link overlap similarity between two actors is 0, as they do not share alter-alter links at all

A visual representation we are proposing is provided in Fig. 1b: shared concepts tend to clump together because links between them have larger ‘attraction power’. Different link weight also causes non-shared concepts repel farther away from the ego. Also notable is that concepts linked with the ‘shared core’ (like ‘ $c5$ ’) are placed closer to the ego than ‘lone-standing’ concepts (like ‘ $c6$ ’).

In what follows we apply this approach to investigate concepts and discuss some particularly interesting concept-centered networks in our data.

3 Data Description

Our data come from an ethnographic study focusing on two local flood management groups located in the County of Shropshire, England. Professional knowledge is represented with a collection of documents issued by flood management agencies and authorities (around 316,000 words in total). Local knowledge is represented with semi-structured interviews with 15 members of the two ‘local flood groups’—local activist groups involved in flood risk management in two villages. We denote these groups as *LFG1* and *LFG2*, respectively. The interviews comprise 186,000 words in total, with the average word number per interview of around 13,000. The total number of co-occurrences for professional network with the threshold of 2 is 266,704 times, while the average total number of co-occurrences in local networks is 790 times.

Note that we represent professional knowledge with one network (further, ‘professional network’). This reflects the ‘universality’ of professional knowledge, as we assume that the content of official documents should reflect some general consensus among professionals. For locals, on the other hand, we allow each interviewee to have their own semantic network. We do so because we are interested in looking at how particular local actors borrow concepts from professional knowledge. We denote local networks with lowercase a’s followed by a number (e.g. *a1* or *a2*).

We produce semantic networks from texts as follows. First, raw texts are tokenized and POS-tagged using the UDpipe package (Wijffels 2019): we convert words into lemmas and combine lemmas with their POS-tags to produce unique concept identifiers (e.g. flood(v) as the verb and flood(n) as the noun), which we refer to as concepts in this paper. Research assistants have manually inspected the corpus checking for machine-missed stopwords (these usually were numbers (e.g. ‘60s’), transcribed artefacts of oral speech (e.g., ‘aha’, ‘eh’), set phrases (‘bear mind’, ‘couple time’), incorrectly automatically recognized words, informants’ real names, and the same words that have different spelling. All such instances have been replaced with either correct versions or a generic placeholder (‘xxx’ sign).

We count co-occurrences using a sliding window approach as they appear within 8-concept vicinity from each other, unless separated by a full stop mark. This yields weighted co-occurrence networks. We then filter these networks from all the non-Nouns, non-Verbs, non-Adjectives as well as from trivial verbs (e.g., ‘do’ or ‘make’), leaving only vertices related to adjectives, nouns, and non trivial verbs. We take this step to reduce the amount of information to process and to focus on informative parts of speech. Finally, we binarize co-occurrence counts using the threshold of 2 for both professional and local networks, that is, we link all the pairs of concepts which co-occurred at least 2 times.

4 The Choice of the Threshold Value

The choice of the threshold value affects topology of the network and, indeed, has an impact on which concepts eventually appear similar in their use by pro-

professionals and locals. In general, extracting binary networks from weighted data is an ongoing research field with several sophisticated methods recently proposed (Dianati 2016; Radicchi et al. 2011; Serrano et al. 2009). We choose to work with the same threshold of 2 for both professional and local networks. Our choice is guided by the following consideration.

Let us consider a hypothetical situation in which the professional network strictly comprises all the local networks. In this case, we would expect all local networks to be sub-networks of the professional network. Let us denote this conjecture as ‘SIN’ (Strictly Included Networks). In this case, we would like to understand how much of the professional network is directly reflected in the local network and why the professional network features many more interlinked concepts than the local networks. Two simple hypothetical scenarios could explain the latter.

In the first scenario, professional knowledge covers more topics than locals use. For instance, professionals may use the concept of *group* in contexts never taken on by locals. Simply put, locals might use a concept in a very narrow context neglecting all the other contexts elaborated in professional knowledge. In this case, the difference in network densities would reflect the breadth of the professional knowledge as opposed to the specificity and particularity of the local knowledge. At the level of network representation, this would imply symmetric thresholding (i.e. if we use 2 for locals, we should use 2 for professionals), because otherwise, we would lose all the contexts in the concept that has been used in the professional knowledge.

The second scenario, at the opposite end of the spectrum, may suggest that local knowledge is just a scaled-down version of the professional knowledge. For example, professionals and locals may have the same local linkage patterns around the concept *group*, but professionals may have more co-occurrences between all the context concepts. Given the SIN hypothesis, we would still argue against removing these richer links.

Finally, higher threshold for the professional network would filter away a great deal of structure to the degree that some local concept neighborhoods would become more linked than the corresponding neighborhoods in the professional network. This goes against our initial assumption that locals draw on professional knowledge. While this well may be the case that there are some concepts more elaborated in the local knowledge than in the professional knowledge, we decided to leave this option for further research and focus instead on the ‘SIN’ hypothesis and its implications.

It is important to note, that our decision to work with the symmetric threshold is driven by the specificity of data: we use two corpora of different sizes. We do not directly engage literature on how to choose such a threshold analytically (for example, Dianati 2016) or what kind of consequences thresholding may have for the topology of the resulting network (Cantwell et al. 2020). However, we are driven by the observation that weight distributions in word co-occurrence networks are typically highly skewed. This implies that a higher threshold value

would discard a large amount of small scale word co-occurrences (Serrano et al. 2009), which, however, might be relevant for local actors.

5 Illustration

We apply our approach to find and examine concepts shared by local activists and professionals. We select concepts for inspection based on their similarity profile, because we want to understand what it means for two semantic networks to share concepts in terms of those concepts' immediate neighbors. Comparing concepts, we take the following steps:

1. First, we create a list of common concepts: those concepts that both professional and local actors use.
2. For each common concept, we extract its immediate alters in the professional and in the local network. This yields concept-centered networks: C_i^{pro} and C_i^{loc}
3. We calculate two metrics characterising similarity between the two concept-centered networks:
 - the vertex overlap
 - the link overlap.

Figure 2 shows vertex and link overlaps for top 15% of the most central concepts in one of the local semantic networks. Let us examine two of them - *management(n)*, *plan(n)*. The concept 'management' is interesting because it yields the highest similarity scores in both vertex and link overlaps. This means that the local actor uses the concept in the same contexts as used in the professional knowledge both in terms of alters and linkages between them. The concept 'plan', on the other hand, has relatively high link overlap but stands out with a relatively low vertex overlap score.

Concept 'management'. One thing to note is that all the contexts in which the local actor uses the concept of 'management' are present in the professional network.

Shared Link 'flood-management'. Let us start with a particular part of this cluster, the link 'flood-management'. The concept of *management* often appears in official documents and local narratives and is likely to play a pivotal role in both professional and local knowledge. There is a general understanding shared by both the locals and professionals that floods cannot be totally eliminated and therefore should be properly managed to minimize their adverse impacts on people and the local economy (Fig. 3).

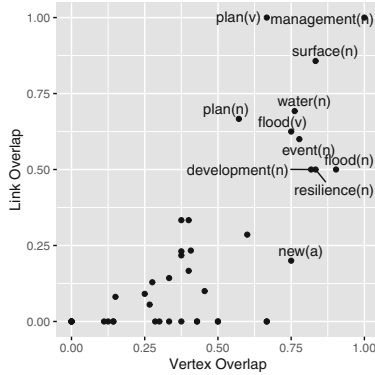


Fig. 2. Top 15% of the most central concepts for local actor 9 along with their vertex and link similarities to professionals. For display purposes, only concepts with link overlap larger than 0.5 or vertex overlap larger than 0.75 are shown

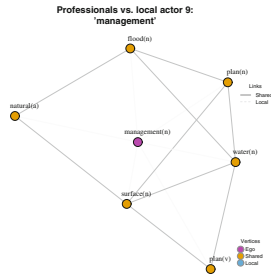


Fig. 3. Professionals vs a local actor: ‘management’. Suffixes after the concept’s name show its part or speech: (n) - noun, (v) - verb, (a) - adjective. For display purposes, only shared and local-specific concepts are shown. In case of ‘management’, the professional network contains 641 other concepts, while the local actor’s network has only 3

Shared Clique ‘management-surface-plan-water’. Professionals often use *surface water management plan* concepts to refer to an official document that coordinates and leads local management practice, with a special aim to minimize flood risk to properties:

*“In 2007 Telford & Wrekin Council were successful in a bid to create a **surface water management plan** under DEFRA s Integrated Urban Drainage pilot studies. The project was driven by the need to gain a better understanding of the surface water environment within its borough with a view to reducing the risk of flooding to existing and new properties through the development control process”* (professional text)

The local actor, meanwhile, points out that *surface water management plan* is a rich source of information on flood risks in the area that informs flood management-related activities of the local flood group:

“Most what I find with most documents related to flooding they’re actually historical reports like the **surface water management plan**, the neighborhood plan, there may have been a report on the Priorslee balancing lake.” (local activist)

Examining the use of ‘management’ in both professional texts and local actor’s interview shows that concept similarity at the level of networks can be traced back to their similarity in texts. Close reading here helps confirming that comparing concept-centered networks at the level of vertex and link overlap indices corresponds to actual similarity of usage:

“the flood manager helped to set up and Jason was quite instrumental in creating the community group and actually gave us an insight into quite important documents like the Surface Water Management Plan or anything else which he could talk to Severn Trent he was quite a good facilitator.” (local activist)

Concept ‘plan’. Figure 4 shows the use of the concept *plan* in the professional and local networks. *Plan* is indeed one of the core concepts in the professional knowledge, in particular, *plan* is embedded into a clique with 4 other concepts some of which also appear in the local actor’s network: *surface-water-flood*. Meanwhile, we can also see that in the local network *plan* has several unique neighbours, most notably inside the dyad *plan-neighborhood* which does not appear in the professional network.

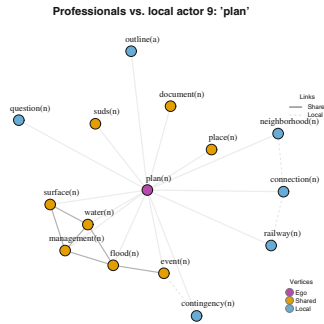


Fig. 4. Professionals vs. local actor 9: ‘plan’. For display purposes, only shared and local-specific concepts are shown. The professional network contains 480 other concepts linked with ‘plan’

Shared Link: Plan-Flood. Professionals and the locals use the concept ‘plan’ when referring to documents that coordinate various stakeholders’ flood management activities. Although both the locals and professionals share the idea of

a document-driven approach to flood management, in practice they may refer to different levels of planning. For example, professionals most often refer to ‘flood risk management plans’ - regional-level documents that orchestrate activities of stakeholders. At the level of semantic network this leads to ‘surface-water-management-plan’ cluster. The regional-level plan, however, does not directly relate to the locals. Speaking about plans, the local actor refers to a ‘neighborhood plan’ - a local document that, among other things, regulates local development to ensure it does not lead to the local drainage system overload. This results in concepts ‘plan’ and ‘neighborhood’ being linked in the local network, yet never appearing next to each other in the professional network. Examples below illustrate this difference in scale for professionals and local activists:

“Flood risk management plans [FRMP] describe the risk of flooding from rivers, the sea, surface water, groundwater and reservoirs. FRMPs set out how risk management authorities will work together and with communities to manage flood and coastal risk over the next 6 years [...] Each EU member country must produce FRMPs as set out in the EU Floods Directive 2007.” (professional texts)

*“I suppose... the other one [issue] which isn’t perhaps as major [a problem] but it [is] certainly significant for [the village], is the local developers. The planning permissions are granted on the understanding that certain flood mitigation steps will be taken... Developers are only allowed to develop in line with the **neighborhood plan**.”* (local activist)

6 Concluding Remarks

This paper proposed a two-step concept-centered approach to compare semantic networks, where one network serves as a “golden standard” from which the other network selectively pulls semantic links. At the first step, we mapped all the shared concepts onto a two-dimensional space of Jaccard similarities of their alters and of links connecting these alters. The joint distribution of these indices highlights concepts which potentially can give insight into the selective appropriation of professional knowledge by local actors. At the second step, we visually inspected chosen concepts using a customized version of the Fruchterman-Reingold layout (Fruchterman and Reingold 1991) which spatially separates shared and non-shared concepts.

We argue that while network comparison can happen at any level of analysis, in the case of semantic networks it is sensible to start with concept-centered networks, since they provide insights on the meaning residing in these networks. We also think that researchers may gain deeper insight into how meaning of concepts in semantic networks emerges because of the productive juxtaposition of quantitative and qualitative perspectives that this vantage brings together.

Future research in concept-centered networks may focus on working with several symmetric thresholds for both professional and local networks. This implies that instead of working with one single network threshold researchers should

embrace multiple network versions and explicitly incorporate this uncertainty into analysis.

Acknowledgement. This work was supported by the Russian Science Foundation (grant 19-18-00394, ‘Creation of knowledge on ecological hazards in Russian and European local communities,’ 2019—ongoing).

References

- Abbott, J.T., Austerweil, J.L., Griffiths, T.L.: Random walks on semantic networks can resemble optimal foraging. *Psychol. Rev.* **122**(3), 558–569 (2015). <https://doi.org/10.1037/a0038693>
- Basov, N., De Nooy, W., Nenko, A.: Local meaning structures: mixed method sociosemantic network analysis. *Am. J. Cult. Sociol.* 1–42 (2019)
- Cantwell, G.T., Liu, Y., Maier, B.F., Schwarze, A.C., Serván, C.A., Snyder, J., St-Onge, G.: Thresholding normally distributed data creates complex networks. *Phys. Rev. E* **101**(6), 062302 (2020)
- Carley, K.: An approach for relating social structure to cognitive structure. *J. Math. Sociol.* **12**(2), 137–189 (1986)
- Carley, K., Newell, A.: The nature of the social agent. *J. Math. Sociol.* **19**(4), 221–262 (1994)
- Choobdar, S., Ribeiro, P., Bugla, S., Silva, F.: Comparison of coauthorship networks across scientific fields using motifs. In: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 147–152. IEEE (2012)
- Dianati, N.: Unwinding the hairball graph: pruning algorithms for weighted complex networks. *Phys. Rev. E* **93**(1), 012304 (2016)
- Fruchterman, T.M., Reingold, E.M.: Graph drawing by force-directed placement. *Softw.: Pract. Exp.* **21**(11), 1129–1164 (1991)
- Lee, M., Marin, J.L.: Coding, counting and cultural cartography. *Am. J. Cult. Sociol.* **3**, 1–33 (2015). <https://doi.org/10.1057/ajcs.2014.13>
- McEwen, L., Jones, O.: Building local/lay flood knowledges into community flood resilience planning after the July 2007 floods, gloucestershire, UK. *Hydrol. Res.* **43**(5), 675–688 (2012)
- Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: simple building blocks of complex networks. *Science* **298**(5594), 824–827 (2002)
- Nye, M., Tapsell, S., Twigger-Ross, C.: New social directions in UK flood risk management: moving towards flood risk citizenship? *J. Flood Risk Manage.* **4**(4), 288–297 (2011)
- Pržulj, N.: Biological network comparison using graphlet degree distribution. *Bioinformatics* **23**(2), e177–e183 (2007)
- Radicchi, F., Ramasco, J.J., Fortunato, S.: Information filtering in complex weighted networks. *Phys. Rev. E* **83**, 046101 (2011). <https://doi.org/10.1103/PhysRevE.83.046101>
- Roth, C., Cointet, J.-P.: Social and semantic coevolution in knowledge networks. *Soc. Netw.* **32**(1), 16–29 (2010)
- Serrano, M. Á., Boguñá, M., Vespignani, A.: Extracting the multiscale backbone of complex weighted networks. *Proc. Natl. Acad. Sci.* **106**(16), 6483–6488 (2009). <https://doi.org/10.1073/pnas.0808904106>. eprint: <https://www.pnas.org/content/106/16/6483.full.pdf>

- Wehn, U., Rusca, M., Evers, J., Lanfranchi, V.: Participation in flood risk management and the potential of citizen observatories: a governance analysis. *Environ. Sci. Pol.* **48**, 225–236 (2015)
- Wijffels, J.: Udpipes: Tokenization, parts of speech tagging, lemmatization and dependency parsing with the ‘udpipe’ ‘nlp’ toolkit. R package version 0.8.3 (2019). <https://CRAN.R-project.org/package=udpipe>

Diffusion and Epidemics



Analyzing the Impact of Geo-Spatial Organization of Real-World Communities on Epidemic Spreading Dynamics

Alexandru Topîrceanu(✉)

Department of Computer and Information Technology,
Politehnica University Timișoara, Timișoara, Romania
alex@cs.upt.ro

Abstract. Models for complex epidemic spreading are an essential tool for predicting both local and global effects of epidemic outbreaks. The ongoing development of the COVID-19 pandemic has shown that many classic compartmental models, like SIR, SIS, SEIR considering *homogeneous mixing* of the population may lead to over-simplified estimations of outbreak duration, amplitude and dynamics (e.g., waves). The issue addressed in this paper focuses on the importance of considering the social organization into geo-spatially organized communities (i.e., the size, position, and density of cities, towns, settlements) which have a profound impact on shaping the dynamics of epidemics. We introduce a novel geo-spatial population model (GPM) which can be tailored to reproduce a similar *heterogeneous* individuals' organization to that of real-world communities in chosen countries. We highlight the important differences between a homogeneous model and GPM in their capability to estimate epidemic outbreak dynamics (e.g., waves), duration and overall coverage using a dataset of the world's nations. Results show that community size and density play an important role in the predictability and controllability of epidemics. Specifically, small and dense community systems can either remain completely isolated, or show rapid bursts of epidemic dynamics; larger systems lengthen the epidemic size and duration proportionally with their number of communities.

Keywords: Computational intelligence · Epidemic spreading · Community structure · Geo-spatial communities · Population model

1 Introduction

Predicting the dynamics of epidemic outbreaks is an important step towards controlling and preventing the spread of infectious diseases. With the recent COVID-19 pandemic affecting most of the world's regions, a lot of scientific effort has been invested in the ability to understand, model and predict the dynamics of the SARS-CoV-2 virus [8]. Similar global efforts have also been registered in the past for the SARS, MERS, Ebola, and even the 1918 flu pandemic, all of which

have helped public health officials prepare better for major outbreaks [10, 19, 20]. As a result, current strategies for controlling and eradicating diseases, including the COVID-19 pandemic, are fueled by consistent insights into the processes that drive, and have driven epidemics in the past [1, 16].

A predominant body of recent research has invested in extending, custom-tailoring, and augmenting standard mass-action mixing models into tools suitable for analyzing COVID-19 [2, 13, 18]. However, in most cases we find that these mathematical models assume homogeneous mixing of the population (i.e., each infected individual has a small chance of spreading infection to every susceptible individual in the population) [2–4, 28]. As such, “flattening the curve”-type solutions have been proposed to decrease the reproduction number R_0 , and thus dampen the peak of the daily infection ratio. Based on homogeneous mixing populations, several notable studies estimate the length and proportion of the current COVID-19 pandemic [2, 4, 13, 17, 18]. By contrast, most pathogens spread through contact networks, such that infection has a much higher probability of spreading to a more limited set of susceptible contacts [15]. Ultimately, governments’ actions around the world are based on these scientific predictions, having immense social and economic impact [3].

Over the past decade, an increasing number of studies pertaining to network science have shown the importance of community structure when considering epidemic processes over networks [5, 6, 23, 26, 27]. In this sense, the heterogeneous organization of communities is not a novel concept in network science [24, 32]. Salathé *et al.* [23] show how community structure affects the dynamics of epidemics, with implications on how networks can be protected from large-scale epidemics. Ghalmane *et al.* [11] reach similar conclusions in the context of time evolving network nodes and edges. Shang *et al.* [26] show that overlapping communities and higher average degree accelerate spreading. In [5] it is shown that overlapping communities lead to a major infection prevalence and a peak of the spread velocity in the early stages of the emerging infection, as the authors Chen *et al.* use a power law model. Stegehuis *et al.* [27] suggest that community structure is an important determinant of the behavior of percolation processes on networks, as community structure can both enforce or inhibit spreading. With a slightly different approach, we find Chung *et al.* [7] who use a multiplex network to model heterogeneity in Singapore’s population; thus, the authors are able to obtain real-world like epidemic dynamics.

Mobility patterns represent an important ingredient for augmenting the realism of complex network models, in order to increase the predictability of epidemic dynamics. Sattenspiel *et al.* [25] incorporate five fixed patterns of mobility into a SIR model to explain a measles epidemic in the Carribean. Salathé *et al.* [24] study US contact networks and conclude that heterogeneity is important because it directly affects the basic reproductive number R_0 , and that it is realistic enough to assume (contact) homogeneity inside communities (e.g., high schools). Their observation supports the simplification of a community’s network, namely from a complex network to a stochastic block model [14]. Finally, Watts *et al.* [32] introduce a synthetic hierarchical block model, capable of reproducing

multiple epidemic waves, but without any correlation to real-world human settlement organization, or realistic distances between communities.

In this paper, we address the issue of modeling mobile heterogeneous population systems, where the community structure is defined by actual real-world geo-spatial data (i.e., position and size of human settlements). The contributions of our study can be summarized as follows:

- We introduce the geo-spatial population model (GPM) to investigate how the duration δ , size ξ and dynamics of an epidemic are quantified, comparing to a similar homogeneous mixing model and to real COVID-19 data [9]. Our research focus is more on the community structure and individual mobility rather than on the transmission model, such that we incorporate GPM into a classic SIR epidemic model to run numerical simulations.
- We define the population system (e.g., a country) as a stochastic block model (SBM) where blocks (or communities) are modeled by real-world settlements from a chosen country. Their size and spatial positioning (latitude, longitude) are set by real-world data.
- We further define original individual mobility patterns based on the population (size) and distance between any pair of communities. Intuitively, individuals are more likely to move to a larger and/or closer settlement, than to a smaller and/or distant one [22].
- We show that the number of settlements in the population system, as well as altering the settlements' density (leading to more compact, or more sparse geo-spatial organization of communities) can directly impact the outbreak duration δ and size ξ .

2 The Geo-Spatial Population Model

We introduce the geo-spatial population model (GPM) as an adaption of the standard stochastic block model [14], where each block, or community, is a human settlement s_i (in a given country or region) characterized by its global position (longitude $x(s_i)$, latitude $y(s_i)$) and number of inhabitants Ω_i (i.e., number of individuals). In this paper, we introduce GPM as a means to model a country's population system, rather than that of the entire planet. Thus, the number and size of all settlements are defined by real-world data for a chosen country. Any individual n_a^i from any settlement s_i is characterized by a stochastic mobility function. The probability $p_a^i(s_j)$ of an individual n_a from settlement s_i to leave to another settlement s_j is given by:

$$p_a^i(s_j) \propto \Omega_j \cdot e^{-d_{ij}/\psi} \quad (1)$$

where Ω_j is the population $|s_j|$ of settlement s_j , d_{ij} is the Euclidean distance $((\Delta x^2 + \Delta y^2)^{1/2})$ between the two settlements s_i and s_j , and ψ (psi) is a tunable parameter. While the Haversine formula is often used in distance calculations over the earth's surface, we consider that distances inside most countries (e.g., tens-hundreds of km) are not affected by the earth's imperfect spherical shape.

Also note that the reference probability of an individual to remain within its settlement s_i , for $d_{ii} = 0$ becomes $p_a^i(s_i) \propto \Omega_i$. In the current form of GPM, all individuals from the same settlement have the same probability for mobility (e.g., $p_a^i(s_j) = p_b^i(s_j), \forall n_a, n_b \in s_i$).

As such, given all probabilities to move from one settlement to all other settlements s_0, \dots, s_k in the population system, we express the normalized probability p^i (summing up to 1, i.e., $\sum_k p^i(s_k) = 1$) by dividing each reference probability from Eq. 1 by the sum of all probabilities for all settlements:

$$p^i(s_j) = \frac{\Omega_j \cdot e^{-d_{ij}/\psi}}{\sum_k \Omega_k \cdot e^{-d_{ik}/\psi}} \quad (2)$$

In practice, we find that the probability of an individual to leave its settlement is roughly 0–2% when the home settlement is moderately large (e.g., a city), and 0–50% when the home settlement is relatively small (e.g., a village or small town). Experimental assessment has shown a reliable value of $\psi = 0.2$ for the tunable parameter; nevertheless, the value of ψ could be customized, in a follow-up study, for each settlement using reliable real mobility data from specific countries [12].

2.1 Real Geo-Spatial Data

An important original contribution of GPM is the fact that it defines stochastic blocks sized and positioned (in a 2D space) based on real geo-spatial data, rather than a synthetic hierarchical construct [32]. As such, we use data from the Global Rural-Urban Mapping Project (GRUMP v1), revision 01 (March 2017) curated by the Center for International Earth Science Information Network (CIESIN), Columbia University [31].

The Grump dataset contains 70,630 entries in *csv* format on human settlements from around the world. The relevant information used by GPM to characterize a community is: country, latitude, longitude, population, name. The dataset is the result of an ongoing large-scale project, as, for example, we find only 24 settlements for Bosnia, totaling 1.1M inhabitants, while the real population is 3.3M (i.e., only 33% of data is available). On the other hand, for Romania, which is a larger country of 19M inhabitants, we find 864 settlements totaling 15.7M inhabitants (83% of data is available). Consequently, we filter out all countries with less than 50 settlements as we consider them incomplete population systems. Additionally, we filter out China, India and the USA because their larger sizes alter the results of the averagely sized countries, and thus deserve separate analysis. The resulting dataset consists of 96 countries (from American Samoa with 32K inhabitants, to Japan with 108M inhabitants).

Figure 1 represents both a conceptual example of computing the GPM mobility probabilities based on position and populations size, as well as a real-world mapping over the Kingdom of Spain. In Fig. 1b, the modeled population is 33M inhabitants (70% of real size) placed in 735 settlements, all within a bounding area of 1000 km \times 850 km (the Canary Islands have been omitted from the figure, but are included in the data model).

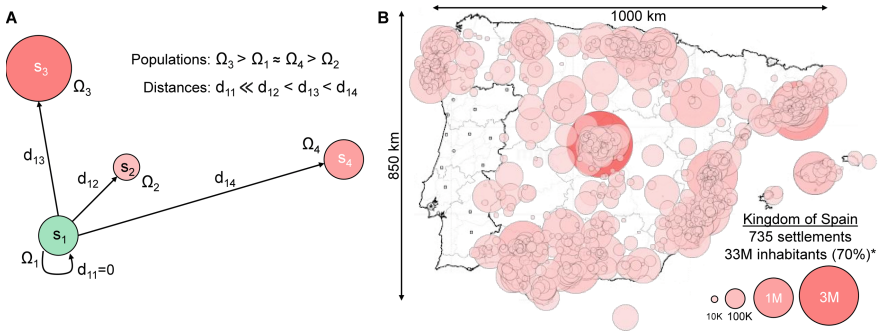


Fig. 1. (A) Conceptual representation of the inter-settlement mobility on an example GPM with 4 settlements s_1 – s_4 . Any individual from s_1 (green) has an associated probability to remain within the same settlement or move to s_2 – s_4 . The parameters affecting the probabilities are: target settlement population Ω_j , and distance d_{ij} to settlement. (B) Example of GPM mapping of 735 settlements over Spain. The total modeled population is 33M, representing $\approx 70\%$ of the real population of Spain (due to dataset incompleteness).

2.2 Epidemic Reference Data

In order to compare our numerical simulations with real epidemic data we use the most recent JHU CSSE COVID-19 dataset curated by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University [9]. The comprehensive dataset contains time series information on daily total confirmed Coronavirus cases for the majority of countries (and some subregions) of the world. From this data we compute the number of new daily cases and show several important insights that are further investigated by our simulations using GPM.

Figure 2a represents the histogram of the current COVID-19 outbreak size around the world. From the bimodal distribution we conclude that many countries are (still) weakly affected by the pandemic (e.g., $\leq 10,000$ total cases), and another significant proportion are strongly affected (e.g., $> 100,000$ cases). In between, there is a relative flat distribution of the outbreak size, similar to the occurrence of measles [32].

In Figs. 2b–d we provide three representative examples of real-world pandemic evolution for the first $\delta \approx 200$ days (starting January 22nd). Here we underline two important empirical observations: (i) the outbreak sizes ($\xi < 1\%$) are much smaller than many early predictions based on homogeneous mixing, (ii) the dynamics are much less predictable, being characterized by multiple waves ($w_{1..3}$) which do not follow a single skewed Gaussian-like wave. Also, the pandemic duration is yet to be accurately inferred from the real data.

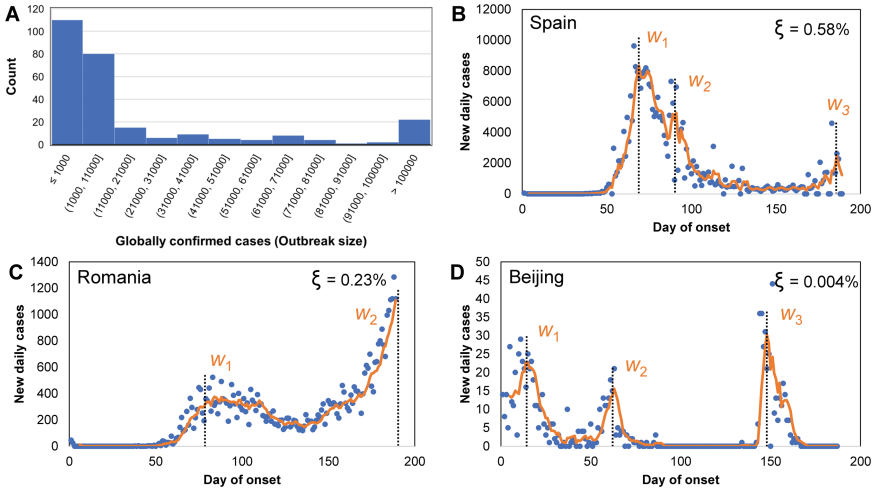


Fig. 2. (A) Histogram of COVID-19 outbreak size, quantified by the number of globally confirmed cases. The current bimodal distribution shows that countries are either weakly or strongly affected by the virus. (B–D) Time series evolution of daily cases from January 22nd to July 26th 2020. The sizes of the current outbreak are given in percent (ξ) and similar several waves ($w_{1..3}$) characterize the COVID-19 epidemic dynamics in varied regions of the world. The current duration of the outbreak is $\delta \approx 200$ days.

3 Results

The numerical simulations running GPM are quantified through the outbreak duration δ and size ξ . All simulations run for a fixed amount of $t = 1000$ iterations, ensuring a 3-year overview of the epidemic. The duration δ represents the number of days (discrete iterations t) from the epidemic onset (iteration $t = 0$) to the last registered new infection case. The size ξ represents the proportion of the total population being infected.

Table 1 offers an overview of the numerical simulations’ statistics on the Grump dataset (before and after filtering out countries with less than 50 settlements). In summary, our simulations do not trigger a pandemic in less than 20 of the smallest countries, a weak pandemic is characteristic to less than 30 countries, and about 20 countries exhibit a strong pandemic (i.e., in terms of size or duration).

Looking at the *persistent* panel in Table 1, we notice that, after leaving just the countries with more than 50 settlements in the dataset, the average duration δ increases, and the average size ξ drops. We believe this is explained by the high number of small-sized countries (101) in which the pandemic may be of shorter duration and higher impact. Furthermore, the top 14 countries (lowest panel) with the longest epidemic duration ($\delta > 270$) take, on average, 446 days to overcome the simulated pandemic, and reach an average infection size of $\xi = 0.65$.

Table 1. Statistics characterizing GPM simulations on the Grump dataset, grouped into four panels: no outbreak was present, weak outbreak (duration $\delta \leq 30$ or size $\xi \leq 0.1$), some moderate outbreak with $\delta > 30$, and a strong outbreak with either $\delta > 270$ (9 months) or $\xi > 0.8$. We measure the number of countries, average number of settlements N_s , population Ω , outbreak duration δ , and outbreak size ξ in each case.

Qualitative	Quantitative	Countries	Avg N_s	Avg Ω	Avg δ	Avg ξ
none	$\delta = 0$	17	35.05	413.17	–	–
none	$\xi = 0$	11	48.72	56.45	–	–
weak	$\delta \leq 30$	29	75.69	4936.91	9.93	0.18
weak	$\xi \leq 0.1$	28	253.82	17916.14	41.78	0.03
persistent	$\delta > 0$	197	191.68	10786.60	81.71	0.53
persistent	$\delta > 0, N_s \geq 50$	96	370.31	20958.42	135.78	0.42
strong	$\xi \geq 0.8$	21	413.57	17115.80	232.14	0.92
strong	$\delta > 270$	14	1232.85	74039.57	446.64	0.65

To compare our heterogeneous mixing GPM with a standard homogeneously mixing model we provide Fig. 3 as an intuitive example. Figures 3a,b show the difference in outbreak size distribution using the same population size (i.e., Spain with 33M inhabitants). In Fig. 3c we extend the measurement to all countries, and the obtained ξ distribution is similar to that on the real COVID-19 data in Fig. 2a. That is, GPM enables realistic outbreak simulations of any size $0 \leq \xi \leq 1$ and duration $\delta \geq 0$. In homogeneously mixing populations, the chances are that the outbreak is short and very strong, or completely nonexistent.

For the homogeneously mixing scenario in Fig. 3d, a representative epidemic trajectory rises rapidly only once, infecting most of the population in the process (here, $\xi = 100\%$). Conversely, in the examples with GPM (Fig. 3e,f), epidemic trajectories exhibit unpredictable rebound, persist for different durations, and may infect different fractions of the population.

The bimodality in the outbreak size distribution further motivates us to analyze the difference between smaller countries (i.e., defined as having less than 50 settlements in the Grump dataset), and larger countries (with more than 50 settlements). In terms of correlation (Pearson ρ) between actual country population and number of settlements in our dataset, we find a $\rho = 0.557$ for ‘smaller’ countries, and $\rho = 0.953$ for ‘larger’ countries. Figure 4a,b plots the outbreak duration and size based on number of settlements for all 101 ‘smaller’ countries (blue) and 96 ‘larger’ countries. We gain two insightful observations:

1. A good distinction is possible between the two groups of countries in terms of outbreak duration (Fig. 4a). Smaller countries’ outbreak duration is shorter and more bounded, within $\delta < 212$ (7 months, $\rho = 0.472$). For larger countries, duration increases as settlements increase ($\rho = 0.668$).

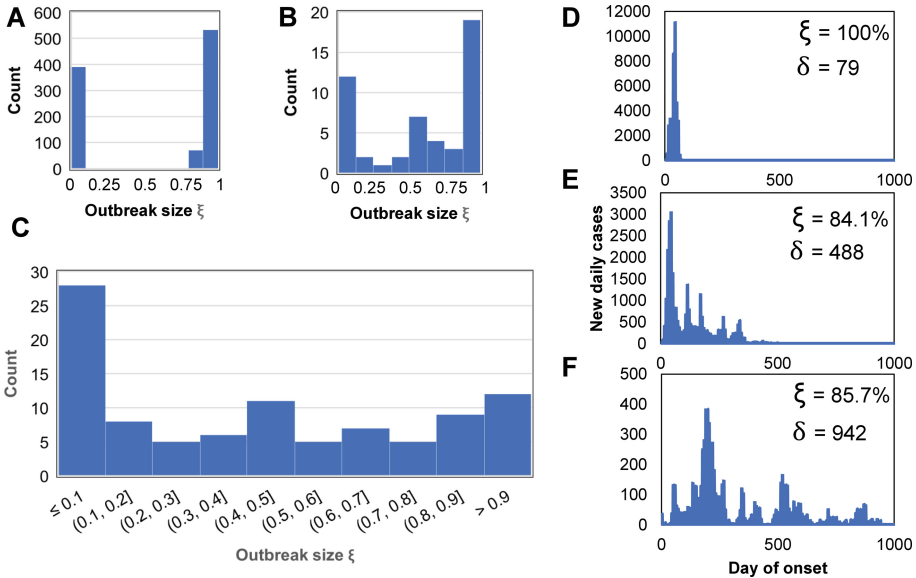


Fig. 3. Evidence for the importance of heterogeneous mixing in simulating epidemic dynamics. (A) Strictly bimodal distribution of size ξ for 1000 simulations of a homogeneously mixing population of Spain (33M). (B) The size ξ distribution on the same population of Spain simulated using GPM. (C) Size ξ distribution of simulated pandemic, on a global scale, comparable to Fig. 2a representing COVID-19 real data. (D–F) Example time series of new daily infection cases using as model the population of Spain (33M individuals). For (D), the population is homogeneously mixing, and for (E–F), the population is structured according to our GPM. (E–F) use the same settings, with $\psi = 0.2$, but exemplify the possibility of a similar sized outbreak with two different durations (488 days vs. 942 days).

2. The same distinction is not possible in terms of outbreak size (Fig. 4b). While larger countries continue to correlate with the number of settlements ($\rho = 0.624$), smaller countries’ outbreak size is unpredictable ($\rho = -0.269$).

Finally, we analyze the impact of settlements density (i.e., similar to controlling the spatial overlapping of communities) in a population system. Starting from the default density ($=1$) given by the actual geographical positioning of settlements, we increase and decrease the population density by three orders of magnitude (i.e., 0.001 to 1000) by contracting/expanding all settlements’ positions proportionally. Figure 4c suggests that only for the default spacing (density ≈ 1) will the outbreak duration be maximized (average $\delta \approx 302$ measured over all countries). In other words, too dense or too sparse environments exhibit short-lived epidemics. To better understand what this means, we provide in Fig. 4d an overview of the outbreak sizes measured over all countries (average $\xi = 0.56$). As such, we find that sparse population systems (density < 1) trigger none or very small short epidemic bursts (e.g., there is not enough population to support the

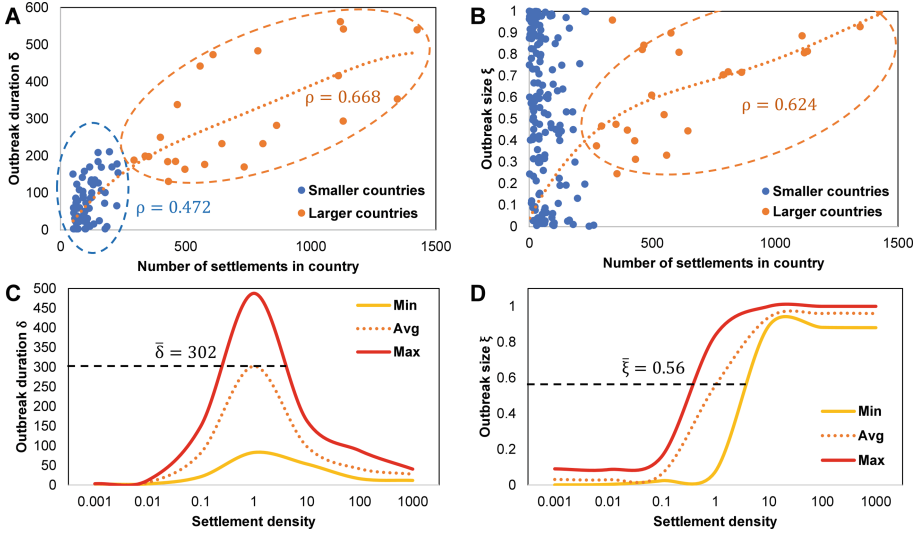


Fig. 4. Dependency between outbreak duration (A), size (B) and number of settlements for smaller (blue) and larger countries (orange). Impact of settlements density on outbreak duration (C) and size (D) when contracting or expanding the position of settlements.

transmission dynamics). Conversely, denser systems (density > 1) also trigger short, but strong epidemics with high coverage. Averagely dense systems present much longer epidemic duration but with a possibly lower size ξ .

4 Conclusions

Establishing realistic models for the geographic spread of epidemics is still underdeveloped compared to other areas of network modeling, such as online social networks [29], models for the diffusion of information [30], or network medicine modeling [21]. Nevertheless, one of the most distinct characteristics of many viral outbreaks is their spreading across geo-spatially organized human communities. In this paper we investigate the importance of spatially structured real-world community structures for predicting epidemic dynamics. The GPM model presented here provides one novel method that may prove useful in better binding complex networks and mathematical epidemics to the empirical patterns of infectious diseases spread across time and space.

Our numerical simulations confirm that smaller scale environments (e.g., countries with fewer settlements) exhibit less predictable epidemic dynamics (in terms of outbreak size ξ), but as a general observation, the duration δ is noticeably shorter (within ≈ 200 days) than that of larger environments. Indeed, for larger environments, the outbreak duration and size increase linearly ($\rho \approx 0.62\text{--}0.67$). In general, our results illustrate the qualitative point

that epidemics, when they succeed, they occur on multiple scales, resulting in longer duration, repeated waves, and hard-to-predict size.

A planned next step in our model is to include diverse isolation measures, under the form of mobility restrictions between settlements, and reduced infectiousness inside settlements (e.g., by wearing masks) and study their feasibility on limiting the infection size on a long term. Furthermore, there are several extensions to GPM worth investigating in future studies. For example, environmental factors associated with settlements location can have important effects on transmission risk, as they can vary greatly over short distances [25]. The model can also consider larger scale populations (e.g., continental) where mobility is given by international travel logs. In-between settlements, real data on mobility patterns can be used when available [12]. Finally, our mobility model may be further detailed to consider contact between individuals along the way to a target settlement (e.g., by car, bus, train) instead of direct transfer (e.g., plane).

Taken together, we believe our model represents a timely contribution to better understanding and tackling the current COVID-19 pandemic that has proven hard to predict with many existing homogeneously mixing population models.

Acknowledgments. This work was supported by a grant of the Romanian Ministry of Education and Research, CNCS - UEFISCDI, project number PN-III-P1-1.1-PD-2019-0379, within PNCDI III.

References

1. Anderson, R.M., May, R.M.: Directly transmitted infections diseases: control by vaccination. *Science* **215**(4536), 1053–1060 (1982)
2. Arenas, A., Cota, W., Gomez-Gardenes, J., Gómez, S., Granell, C., Matamalas, J.T., Soriano-Panos, D., Steinegger, B.: A mathematical model for the spatiotemporal epidemic spreading of covid19. *MedRxiv* (2020)
3. Atkeson, A.: What will be the economic impact of covid-19 in the us? Rough estimates of disease scenarios. Technical Report, Nat. Bureau of Economic Research (2020)
4. Block, P., Hoffman, M., Raabe, I.J., Dowd, J.B., Rahal, C., Kashyap, R., Mills, M.C.: Social network-based distancing strategies to flatten the covid-19 curve in a post-lockdown world. *Nat. Hum. Behav.* **4**, 1–9 (2020)
5. Chen, J., Zhang, H., Guan, Z.H., Li, T.: Epidemic spreading on networks with overlapping community structure. *Phys. A: Stat. Mech. Appl.* **391**(4), 1848–1854 (2012)
6. Cherifi, H., Palla, G., Szymanski, B.K., Lu, X.: On community structure in complex networks: challenges and opportunities. *Appl. Netw. Sci.* **4**(1), 1–35 (2019)
7. Chung, N.N., Chew, L.Y.: Modelling singapore covid-19 pandemic with a seir multiplex network model. *medRxiv* (2020)
8. Cohen, J., Kupferschmidt, K.: Labs scramble to produce new coronavirus diagnostics (2020)
9. Dong, E., Du, H., Gardner, L.: An interactive web-based dashboard to track covid-19 in real time. *Lancet Infect. Dis.* **20**(5), 533–534 (2020)

10. Dye, C., Gay, N.: Modeling the SARS epidemic. *Science* **300**(5627), 1884–1885 (2003)
11. Ghalmane, Z., Cherifi, C., Cherifi, H., El Hassouni, M.: Centrality in complex networks with overlapping community structure. *Sci. Rep.* **9**(1), 1–29 (2019)
12. Hasan, S., Zhan, X., Ukkusuri, S.V.: Understanding urban human activity and mobility patterns using large-scale location-based data from online social media. In: *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing*, pp. 1–8 (2013)
13. Hellewell, J., Abbott, S., et al.: Feasibility of controlling covid-19 outbreaks by isolation of cases and contacts. *Lancet Global Health* **8**(4), e488–e496 (2020)
14. Holland, P.W., Laskey, K.B., Leinhardt, S.: Stochastic blockmodels: first steps. *Soc. Netw.* **5**(2), 109–137 (1983)
15. Keeling, M.: The implications of network structure for epidemic dynamics. *Theor. popul. Biol.* **67**(1), 1–8 (2005)
16. Keeling, M.J., Rohani, P.: *Modeling Infectious Diseases in Humans and Animals*. Princeton University Press, Princeton (2011)
17. Koo, J., Cook, A., Park, M., et al.: Interventions to mitigate early spread of covid-19 in Singapore: a modelling study. *Lancet Infect. Dis.* (2020)
18. Kucharski, A.J., Russell, T.W., et al.: Early dynamics of transmission and control of covid-19: a mathematical modelling study. *Lancet Infect. Dis.* **20**(5), 553–558 (2020)
19. Lipsitch, M., Cohen, T., et al.: Transmission dynamics and control of severe acute respiratory syndrome. *Science* **300**(5627), 1966–1970 (2003)
20. Lloyd-Smith, J.O., Schreiber, S.J., et al.: Superspreading and the effect of individual variation on disease emergence. *Nature* **438**(7066), 355–359 (2005)
21. Mihaicuta, S., Udrescu, M., Topirceanu, A., Udrescu, L.: Network science meets respiratory medicine for OSAS phenotyping and severity prediction. *Peer J.* **5**, e3289 (2017)
22. Nguyen, A.D., Sénac, P., Ramiro, V., Diaz, M.: Steps-an approach for human mobility modeling. In: *International Conference on Research in Networking*, pp. 254–265. Springer (2011)
23. Salathé, M., Jones, J.H.: Dynamics and control of diseases in networks with community structure. *PLoS Comput. Biol.* **6**(4), e1000736 (2010)
24. Salathé, M., Kazandjieva, M., Lee, J.W., Levis, P., Feldman, M.W., Jones, J.H.: A high-resolution human contact network for infectious disease transmission. *Proc. Nat. Acad. Sci.* **107**(51), 22020–22025 (2010)
25. Sattenspiel, L., Dietz, K., et al.: A structured epidemic model incorporating geographic mobility among regions. *Math. Biosci.* **128**(1), 71–92 (1995)
26. Shang, J., Liu, L., Li, X., Xie, F., Wu, C.: Epidemic spreading on complex networks with overlapping and non-overlapping community structure. *Phys. A: Stat. Mech. Appl.* **419**, 171–182 (2015)
27. Stegehuis, C., Van Der Hofstad, R., Van Leeuwen, J.S.: Epidemic spreading on complex networks with community structures. *Sci. Rep.* **6**(1), 1–7 (2016)
28. Thunström, L., Newbold, S.C., Finnoff, D., Ashworth, M., Shogren, J.F.: The benefits and costs of using social distancing to flatten the curve for covid-19. *J. Benefit-Cost Anal.* **11**(2), 1–27 (2020)
29. Topirceanu, A., Udrescu, M., Vladutiu, M.: Genetically optimized realistic social network topology inspired by facebook. In: *Online Social Media Analysis and Visualization*, pp. 163–179. Springer (2014)

30. Topirceanu, A., Udrescu, M., Vladutiu, M., Marculescu, R.: Tolerance-based interaction: a new model targeting opinion formation and diffusion in social networks. *Peer J. Comput. Sci.* **2**, e42 (2016)
31. Warszawski, L., Frieler, K., et al.: Center for international earth science information network—ciesin—columbia university. gridded population of the world, version 4 (gpwv4). NASA socioeconomic data and applications center (sedac), Atlas of Environmental Risks Facing China Under Climate Change, p. 228 (2017). <https://doi.org/10.7927/h4np22dq>
32. Watts, D.J., Muhamad, R., Medina, D.C., Dodds, P.S.: Multiscale, resurgent epidemics in a hierarchical metapopulation model. *Proc. Nat. Acad. Sci.* **102**(32), 11157–11162 (2005)



Identifying Biomarkers for Important Nodes in Networks of Sexual and Drug Activity

Jacob Grubb, Derek Lopez, Bhuvaneshwar Mohan, and John Matta^(✉)

Southern Illinois University Edwardsville, Edwardsville, IL 62025, USA
jmatta@siue.edu

Abstract. This paper uses network science techniques to evaluate the SATHCAP dataset concerning HIV and drug use. A referral network is generated via respondent-driven sampling, which is used to identify important bridge nodes that are responsible for maintaining the structure of large connected components of sexual and drug-using activity. These nodes are scrutinized to determine biomarkers and social factors that distinguish them from the underlying population. It is found that attributes such as homelessness and sexual abuse are more prevalent in these bridge nodes. These nodes are ill-served by public health efforts, because they are hard to reach and difficult to identify. Intervention campaigns targeted at groups displaying these attributes could meaningfully lower the spread of HIV.

Keywords: Network theory · Data mining · Public healthcare

1 Introduction

Problems associated with HIV transmission extend into all aspects of society, with minority populations often bearing the brunt in both health and economic outcomes. For example, 1 in 7 people with HIV does not know it, one in two minority men who have sex with men (MSM) will become HIV-infected in their lifetime, and African American women bear a disproportionate HIV burden and poorer health outcomes than other women [23].

A critical barrier to the eradication of HIV is identification, education, and treatment of infected and potentially infected individuals. If interventions are to be successful, they must be targeted on segments of individuals who play important roles in HIV transmission [20]. The effectiveness of highly-targeted medical interventions has been demonstrated in studies such as [5] in which no examples of heterosexual HIV transmission were found when an HIV-positive partner was receiving HAART.

Based on work in [3] and [24], it is thought that transmission and substance-use networks consist of tree-like sub-groups joined by a central cycle. This study

concentrates on nodes with high betweenness centrality, who are important participants because of their topological position on the central cycle of the transmission network. Epidemiological theory suggests that interventions targeting these nodes are more effective in stopping the spread of a disease through the network than interventions involving other nodes [9,18].

This study is a secondary analysis of SATHCAP (Sexual Acquisition and Transmission of HIV Cooperative Agreement Program) [15] user data for Chicago, Los Angeles, and Raleigh. The SATHCAP data was collected as part of a study to assess the impact of drug use in the sexual transmission of HIV from traditional high-risk groups to lower risk groups. The SATHCAP dataset was collected by a process called *Respondent-Driven Sampling* (RDS). This process naturally leads to a network-oriented presentation of the data. Although there are known theoretical benefits to statistical analysis with RDS data [16,25], there is some question as to whether the dataset can be accurately analyzed using classical statistical techniques. An evaluation of the statistical implications for respondent-driven sampling can be found in Lee et al. [17]. They conclude that RDS results in a non-random sampling process with errors that make traditional statistical analysis unwieldy and ineffective. This dissatisfaction with traditional statistical analysis has led to underuse of the datasets, as well as underuse of the RDS technique.

The data have been obtained through the National Addiction and HIV Data Archive Program (NAHDAP), accessible online¹. This research was conducted under the approval of the Southern Illinois University Edwardsville IRB.

2 Related Work

There have been several successful studies on finding important nodes using the SATHCAP data, including a special issue of the Journal of Urban Health in 2009 [6]. In Youm et al. [27] neighborhoods are identified in Chicago for localized campaigns. The neighborhoods are hidden because they have fewer than average cases of HIV. However, they are important in transmission, as they act as bridge neighborhoods facilitating spreading. Simple factors can cause individuals to be important, yet overlooked, such as being very poor [22], or ethnic [28].

There are difficulties with using RDS data for statistical analysis, which is intended to be performed on randomly collected data. It is shown in [11] that the length of data collection chains is often insufficient to obtain an unbiased sample, and it is stated in [12] that the poor statistical “performance of RDS is driven not by the bias but by the high variance of estimates.” It is shown in [26] that valid point estimates are possible with RDS data, but that improvement in variance estimation is needed. The current study avoids this controversy by using network science techniques to analyze RDS data.

The graphical nature of respondent driven sampling is examined in [7]. Betweenness centrality is a widely-used concept for finding nodes important to

¹ <https://www.icpsr.umich.edu/icpsrweb/NAHDAP/index.jsp>.

network resilience and spreading [14], including HIV transmission [4]. Important nodes in other epidemiological contexts have been called superspreaders [19].

3 Methodology

3.1 Data Acquisition and Curation

SATHCAP Origins. The SATHCAP study was conducted across three cities in the United States: Chicago, Los Angeles, and the Raleigh-Durham area, as well as St. Petersburg, Russia [15]. This study used a system of peer-recruitment and respondent-driven sampling to generate a data sampling of men who have sex with men (MSM), drug users (DU), and injected drug users (IDU).

The peer recruitment process allowed individuals to recruit sexual and drug partners to participate in the survey, and for those partners to recruit additional partners. The participants were provided a set of colored coupons for referring partners to the survey, with different colors representing different relationships, such as male sexual partnership, female sexual partnership, or high risk behaviour (MSM, DU, and IDU). Participants in the program were asked a series of questions that attempted to gather as much information about the participant’s sexual and drug activity as possible.

SATHCAP Network Conversion. The process of converting the SATHCAP dataset into a network format was accomplished through the use of Python data science tools, including PANDAS [21] and NetworkX [13]. The network generated by this dataset represents the recruitment of partners within the survey, with nodes being the individual participants and edges representing recruitment. The data was accumulated by iterating through each response to the survey and generating an edge between two respondents when a distributed coupon number matched a respondent coupon number within the survey.

The overall recruitment network consists of 4688 nodes and 4276 edges, with a total of 412 connected components. A majority (255/412) of these components are trivial graphs of size one and two, representing a case where a “seed” participant recruited either one additional participant or was unable to recruit any additional participants to the program. A further number (125/412) of components are of size 3–30, representing a network of referrals that showed limited success, where multiple rounds of recruitment occurred, but failed to spread significantly. The largest (32/412) components range in size from 30 nodes to 949 nodes. These large components represent the successful chains of recruitment intended by the program. Every connected component within the referral network demonstrates a classical tree structure [8]. Every node can be recruited at most one time and can recruit up to 6 other nodes. We divided the network into sub-graphs based on the city in which the respondent took the survey. Statistics including number of nodes, edges, and components can be found in Table 1.

Feature Reduction. The original SATHCAP dataset contains a total of 1488 features and 4688 observations. Many of these features ($n = 1352/1488$) are missing more than 40% of observations. We experimentally resized the number of features based on the number of missing observations and determined that an optimal setting was to retain all features containing at least 94% of observations. Eight of the features were metadata and were also removed. This reduction converted our dataset to 80 features and 4688 observations.

Table 1. Basic Statistics about the individual city sub-forests

City	# Nodes	# Edges	# Components	Largest comp.
Chicago	2739	2607	132	949
Los Angeles	845	728	117	100
Raleigh	1104	941	163	139

Data Normalization and One-Hot Encoding. Features containing multi-value attributes were processed using one-hot encoding. For example, categorical data such as *site where survey was taken* were converted to Yes-No variables such as *site-Chicago*, *site-LosAngeles*, and *site-Raleigh*. The data was then normalized using the min-max scaling method to yield values between 0 and 1. This process resulted in a total of 143 features once all remaining meta-data was removed.

3.2 Calculating Betweenness Centrality

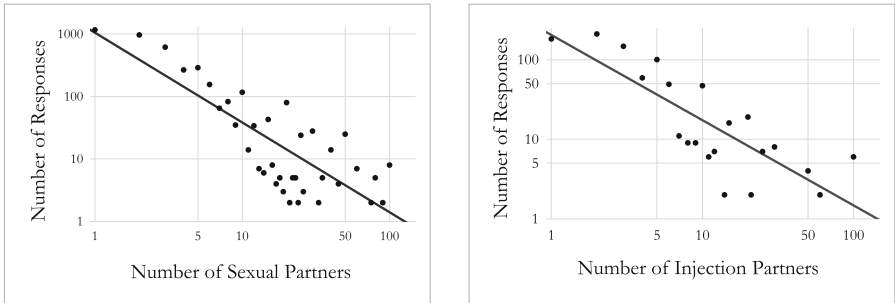
The betweenness centrality of a node is a powerful measure of that node’s overall influence [10] within a network. The betweenness centrality of a node is defined to be the number of shortest paths that node lies upon, with respect to the total number of shortest paths possible within the network. Here a shortest path between nodes s and t is defined as the path from s to t containing the fewest hops. A mathematical formula describing this measure can be seen in Eq. 1, where σ_{st} represents the total number of shortest paths from node s to node t and $\sigma_{st}(v)$ represents the number of those paths that contain node v .

$$b(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \quad (1)$$

Using this definition, we calculated the exact betweenness centrality of every node within the SATHCAP referral network. Naive computation of betweenness centrality on an unweighted graph has a $O(|V||E|)$ time complexity.

3.3 Correlation of Features with High Betweenness

For each city network, we identified the 10 nodes with the highest betweenness centrality. For each of these nodes, we compared each of the 143 features with the average of the underlying city population. A feature was marked as notable for the individual if the value was more than two standard deviations away from the underlying city average. By accumulating the number of notable features across the top 10 highest betweenness nodes, patterns began to emerge across features and cities.



(a) Distribution of reported number of sexual partners within the last six months

(b) Distribution of reported number of injected drug partners within the last six months

Fig. 1. Distribution of underlying sexual and drug networks, plotted on a log-log scale. The downward slope is characteristic of a scale-free network.

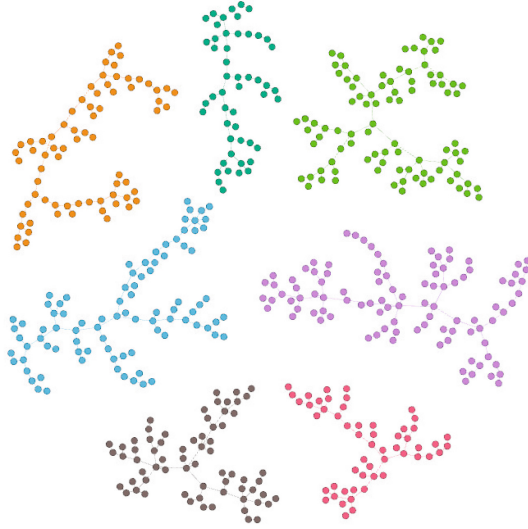
4 Results

4.1 Scale-Free Underlying Networks

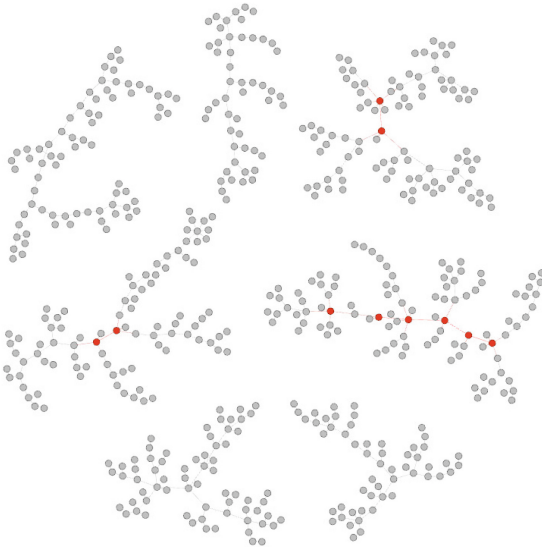
The degree distributions of many real-world networks follow a power-law. This type of network is commonly referred to as *scale-free* [2]. When the degrees of the nodes, x , are plotted against the number of occurrences, $f(x)$, within the network on a log-log chart, the trendline can be described by an equation such as shown in Eq. 2.

$$f(x) = x^{-\gamma} \quad (2)$$

In the case of SATHCAP, we find that the underlying networks of sexual and drug activity can be considered scale-free, based upon the distribution of values within the variables describing the number of unique sexual partners the respondent has had during the past six months (shown in Fig. 1(a)) and the number of unique people with whom the respondent has injected drugs within the past six months (shown in Fig. 1(b)). Figures 1(a) and 1(b) are plotted on a



(a) Connected Components of the Los Angeles Referral network with more than 30 nodes. Different colors indicate unique components of the network.



(b) Connected Components of the Los Angeles Referral network with more than 30 nodes. The red nodes indicate the top 10 highest betweenness centralities within the city.

Fig. 2. Largest components of the Los Angeles SATHCAP network.

log-log scale, and show the distinct downward slope representative of a power-law degree distribution, indicating a scale-free network.

This is an important result, because it is well known that scale-free networks demonstrate resilience to random attacks but high susceptibility to targeted attacks [1]. The scale-free property of networks indicates the existence of important nodes that have a greater effect on the overall structure of the graph. Therefore, targeting important nodes within the referral network is a reasonable strategy on which to base efforts to stop spreading.

4.2 City Graphs with High Betweenness Nodes

A graphical representation of the Los Angeles referral network is shown in Fig. 2. Figure 2(a) shows the Los Angeles Referral Graphs colored by component. For the sake of readability, the graphs show only the components containing more than 30 nodes. There exist many smaller components within each city, in addition to the components shown. Using the definition of betweenness centrality provided in Eq. 1, we identified the top 10 nodes with highest betweenness centrality in each city graph. Subfigure 2(b) shows the highest betweenness centrality nodes of Los Angeles highlighted in red.

Interestingly, our first expectations for the high betweenness nodes were that they would contain the “seed” nodes of the referral system. We expected that despite the randomness introduced by recruitment, the RDS system would average out to be a roughly balanced tree with the seed nodes as the root and therefore the highest betweenness nodes. This was not the case. Within all three cities, none of the high betweenness nodes were seeds within the referral program. The high betweenness nodes therefore must be indicating something different than the location or order in which they were recruited.

4.3 Exceptional Attributes per City

We compared the value of the attributes of the high betweenness nodes within each city to the average value across the entire city. We consider an attribute of a high betweenness node to be *exceptional* if it is more than two standard deviations away from the underlying city average for that attribute. The exceptionality of an attribute is calculated as the number of times a high betweenness node indicated that attribute to be exceptional divided by the total number of high betweenness nodes for that city, in this case 10. We performed this calculation across all 3 cities (shown in Table 2), then accumulated all exceptionality into a single score across all three cities (shown in Table 3).

Table 2 shows that certain attributes are exceptional across high betweenness nodes in all three cities. In particular, the attribute “slept-2” or an indication that the respondent most often slept in their neighborhood, but not in their own home was prevalent across all cities and was distinctly higher in high betweenness nodes than within the underlying population. In the Raleigh-Durham data, three of 5 top attributes had to do with living arrangements, as did two with Los Angeles and one with Chicago. Other attributes such as “usede” which indicates

Table 2. Exceptional Attributes per City

RALEIGH-DURHAM		
Attribute	Plaintext	Exceptionality
slept-2	Last week, I most often slept in my neighborhood, but not my home.	0.4
tmode-5	My primary form of transportation is walking.	0.3
usedc	I have used heroin and cocaine mixed together (speedball).	0.3
reside-3	I currently live in a lover's apartment or house.	0.2
slept-3	Last week, I most often slept in a different neighborhood within 20 miles of my home.	0.2

CHICAGO		
Attribute	Plaintext	Exceptionality
slept-2	Last week, I most often slept in my neighborhood, but not my home.	0.3
mstat-5	I am currently divorced.	0.3
risk2	My first sexual encounter was non-consensual.	0.3
insd2	I have experienced difficulty getting healthcare due to my race/ethnicity.	0.2
insd4	I have experienced difficulty getting healthcare due to my culture/heritage.	0.2

LOS ANGELES		
Attribute	Plaintext	Exceptionality
usedi	Drug Usage (other)	0.5
reside-5	I currently live in a rented room at a hotel or a rooming house	0.4
slept-2	Last week, I most often slept in my neighborhood but not my home.	0.3
sexid2-4	I mostly have sex with women, but occasionally men	0.3
racee	Other Race	0.3

that the respondent used a mixture of cocaine and heroin are more prevalent within Raleigh-Durham, but not as common in Chicago or Los Angeles, although the generic category of “Drug Usage (other)” had the highest exceptionality in Los Angeles. Beyond living arrangements and drug use, difficulty getting

healthcare and having sex with both men and women were represented among the most exceptional attributes.

Table 3. Exceptional Attributes of the overall SATHCAP network

Attribute	Plaintext	Exceptionality
slept-2	Last week, I most often slept in my neighborhood, but not my home	0.3333
mstat-5	I am currently divorced	0.2
usedi	Drug Usage (other)	0.2
reside-5	I currently live in a rented room at a hotel or a rooming house	0.1667
risk1	Age of first sexual encounter	0.1667
risk2	My first sexual encounter was non-consensual	0.1667
sexid2-2	I have sex mostly with men, but occasionally with women	0.1667
tmode-5	My primary form of transportation is walking	0.1333
insd5	I have experienced difficulty getting healthcare due to my alcohol/drug use	0.1333
insd2	I have experienced difficulty getting healthcare due to my race/ethnicity	0.1

As shown in Table 3, certain categories of attributes were more common within high betweenness nodes than the overall city averages. Attributes such as “slept-2”, “reside-5”, and “tmode-5” indicate the respondent is more prone to homelessness, while attributes such as “risk1” and “risk2” indicate a history of sexual abuse. Drug use and having sex with both men and women also have high exceptionality within the overall network.

4.4 Unique Attributes of High Betweenness Nodes

As seen in Fig. 2, many of the high betweenness nodes share a close proximity to one another within the referral network, indicating a shared subset of attributes. Nodes with high betweenness centrality are often referred to as “bridge” nodes because they connect, or bridge, different groups. Conversely, this suggests that while some attributes are shared among the high betweenness nodes, other attributes would be unique. Figure 3 shows the largest component in Los Angeles, which contains 60% of that city’s high betweenness nodes. By analyzing each node individually, we discovered each node had a set of unique attributes that no other high betweenness node in that city contained.

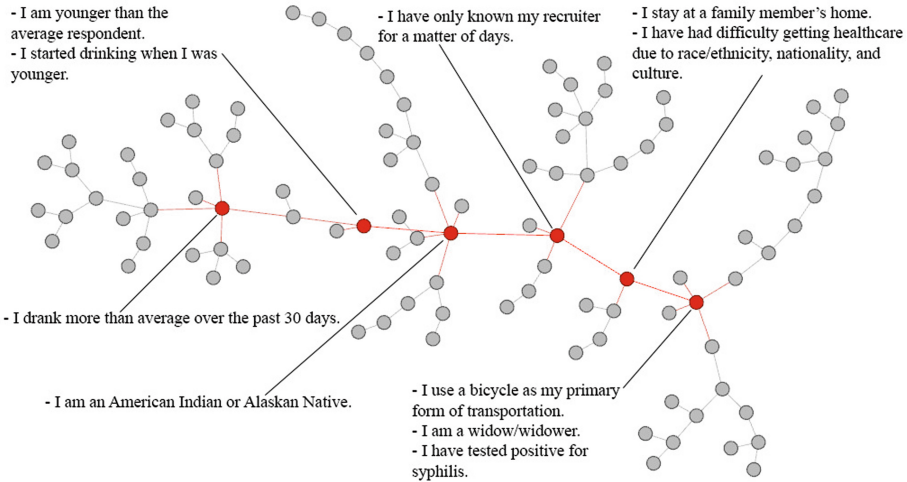


Fig. 3. The largest connected component in Los Angeles. The nodes with the six highest betweenness centrality scores are shown in red. Unique exceptional attributes are shown for each node.

The exceptional attributes shown in Fig. 3 are unique to each of the associated respondents within the top 10 high betweenness nodes. The descriptive qualities of these attributes provide a unique insight into the structure of these networks of recruitment, via outlier data that may not be captured by traditional statistical modeling. By focusing on the centrality “bridging” qualities of these nodes, we can identify core attributes that may not otherwise be captured. For example, attributes such as youth, being a member of an underrepresented race, and/or carrying STIs are identifiable as attributes that bridge from one group to another.

5 Discussion and Conclusion

We look to complex network theory to provide additional tools for the management and prevention of the spread of disease. We use the SATHCAP Respondent-Driven Sampling (RDS) dataset as the basis for a methodology of betweenness-centrality based biomarker discovery.

By analyzing the coupon-code based distribution system of SATHCAP, we were able to create a complex forest of tree structures across three large US cities: Chicago, Los Angeles, and Raleigh-Durham. We used the responses themselves as nodes and drew edges between nodes to represent lines of recruitment. We found nontrivial connected components within each city. These trees represent large, successful chains of recruitment by RDS, and are a subset of the underlying social network, from which we are able to draw relevant conclusions.

We find that the underlying social networks of sexual activity and concurrent drug usage follow a power-law distribution, a key indicator of a scale-free relationship. With consideration that the SATHCAP network is a subset of these

social networks, we wanted to identify a set of highly influential nodes within the referral network. We used betweenness centrality as a measure of a node's influence on the structure and connectivity of these graphs. With the highly influential nodes identified, we attempted to find a set of attributes that distinguish these influential nodes from the underlying groups of individuals. By looking at the average values for each attribute and comparing them to the individual values of each high betweenness node, we were able to identify exceptional attributes that fell more than two standard deviations away from the population average. By accumulating these exceptional attributes across all high betweenness nodes and across all cities, we found sets of attributes that are shared as well as sets that were unique within these influential nodes.

The set of attributes associated with high betweenness indicate two major themes, homelessness and sexual abuse. Individuals with high betweenness have higher rates of attributes commonly associated with homelessness, such as living in rented rooms, sleeping around a home neighborhood, and walking as a primary form of transportation. High betweenness individuals are more likely to have had non-consensual sexual encounters as well as sexual encounters occurring earlier in life, two indicators of sexual abuse. Individual examination revealed unique attributes within the high betweenness nodes such as heavy alcohol use, STIs, and being a member of an underrepresented race. These unique attributes show the broad range of targets towards which interventions could be directed.

The existence of higher levels of homelessness, sexual abuse, drug use, and pansexuality within high betweenness nodes indicates that these demographics are prime candidates for a targeted intervention program. The removal of these nodes from the network through social programs or educational interventions would have a distinct impact on the overall structure of the underlying network of drug use and sexual activity, limiting the spread of HIV, sexual and drug-use transmitted diseases.

References

1. Albert, R., Jeong, H., Barabási, A.L.: Error and attack tolerance of complex networks. *Nature* **406**(6794), 378–382 (2000)
2. Barabási, A.L., Bonabeau, E.: Scale-free networks. *Sci. Am.* **288**(5), 60–69 (2003)
3. Bearman, P.S., Moody, J., Stovel, K.: Chains of affection: the structure of adolescent romantic and sexual networks. *Am. J. Sociol.* **110**(1), 44–91 (2004)
4. Bell, D.C., Atkinson, J.S., Carlson, J.W.: Centrality measures for disease transmission networks. *Soc. Netw.* **21**(1), 1–21 (1999)
5. Castilla, J., Del Romero, J., Hernando, V., Marincovich, B., García, S., Rodríguez, C.: Effectiveness of highly active antiretroviral therapy in reducing heterosexual transmission of HIV. *JAIDS J. Acquired Immune Defic. Syndr.* **40**(1), 96–101 (2005)
6. Compton, W., Normand, J., Lambert, E.: Sexual acquisition and transmission of HIV cooperative agreement program (SATHCAP), July 2009. *J. Urban Health* **86**(1), 1–4 (2009)
7. Crawford, F.W.: The graphical structure of respondent-driven sampling. *Sociol. Methodol.* **46**(1), 187–211 (2016)

8. Deo, N.: Graph theory with application to engineering and computer science, pp. 39–44. phi pvt., Ltd., India (1974)
9. Eames, K.T.: Modelling disease spread through random and regular contacts in clustered populations. *Theor. popul. Biol.* **73**(1), 104–111 (2008)
10. Freeman, L.C.: A set of measures of centrality based on betweenness. *Sociometry* **40**, 35–41 (1977)
11. Gile, K.J., Handcock, M.S.: 7. respondent-driven sampling: an assessment of current methodology. *Sociol. Methodol.* **40**(1), 285–327 (2010)
12. Goel, S., Salganik, M.J.: Assessing respondent-driven sampling. *Proc. Nat. Acad. Sci.* **107**(15), 6743–6747 (2010)
13. Hagberg, A., Swart, P., S Chult, D.: Exploring network structure, dynamics, and function using networkx. Technical Report, Los Alamos National Lab. (LANL), Los Alamos, NM (United States) (2008)
14. Holme, P., Kim, B.J., Yoon, C.N., Han, S.K.: Attack vulnerability of complex networks. *Phys. Rev.* **65**(5), 056109 (2002)
15. Iguchi, M.Y., Ober, A.J., Berry, S.H., Fain, T., Heckathorn, D.D., Gorbach, P.M., Heimer, R., Kozlov, A., Ouellet, L.J., Shoptaw, S., et al.: Simultaneous recruitment of drug users and men who have sex with men in the united states and Russia using respondent-driven sampling: sampling methods and implications. *J. Urban Health* **86**(1), 5 (2009)
16. Kuhns, L.M., Kwon, S., Ryan, D.T., Garofalo, R., Phillips, G., Mustanski, B.S.: Evaluation of respondent-driven sampling in a study of urban young men who have sex with men. *J. Urban Health* **92**(1), 151–167 (2015)
17. Lee, S., Suzer-Gurtekin, T., Wagner, J., Valliant, R.: Total survey error and respondent driven sampling: focus on nonresponse and measurement errors in the recruitment process and the network size reports and implications for inferences. *J. Official Stat.* **33**(2), 335–366 (2017)
18. Lloyd, A.L., May, R.M.: How viruses spread among computers and people. *Science* **292**(5520), 1316–1317 (2001)
19. Lloyd-Smith, J.O., Schreiber, S.J., Kopp, P.E., Getz, W.M.: Superspreading and the effect of individual variation on disease emergence. *Nature* **438**(7066), 355–359 (2005)
20. Magnani, R., Sabin, K., Saidel, T., Heckathorn, D.: Review of sampling hard-to-reach and hidden populations for HIV surveillance. *Aids* **19**, S67–S72 (2005)
21. McKinney, W.: Data structures for statistical computing in python. In: van der Walt, S., Millman, J. (eds.) *Proceedings of the 9th Python in Science Conference*, pp. 51 – 56 (2010)
22. Murphy, R.D., Gorbach, P.M., Weiss, R.E., Hucks-Ortiz, C., Shoptaw, S.J.: Seroadaptation in a sample of very poor Los Angeles area men who have sex with men. *AIDS Behav.* **17**(5), 1862–1872 (2013)
23. Pellowski, J.A., Kalichman, S.C., Matthews, K.A., Adler, N.: A pandemic of the poor: social disadvantage and the us HIV epidemic. *Am. Psychol.* **68**(4), 197 (2013)
24. Potterat, J.J., Phillips-Plummer, L., Muth, S.Q., Rothenberg, R., Woodhouse, D., Maldonado-Long, T., Zimmerman, H., Muth, J.: Risk network structure in the early epidemic phase of HIV transmission in Colorado springs. *Sex. Transm. Infect.* **78**(suppl 1), i159–i163 (2002)
25. Rhodes, S.D., McCoy, T.P.: Condom use among immigrant Latino sexual minorities: multilevel analysis after respondent-driven sampling. *AIDS Educ. Prev.* **27**(1), 27–43 (2015)

26. Wejnert, C.: 3. an empirical test of respondent-driven sampling: point estimates, variance, degree measures, and out-of-equilibrium data. *Sociol. Methodol.* **39**(1), 73–116 (2009)
27. Youm, Y., Mackesy-Amiti, M.E., Williams, C.T., Ouellet, L.J.: Identifying hidden sexual bridging communities in Chicago. *J. Urban Health* **86**(1), 107–120 (2009)
28. Young, S.D., Shoptaw, S., Weiss, R.E., Munjas, B., Gorbach, P.M.: Predictors of unrecognized HIV infection among poor and ethnic men who have sex with men in Los Angeles. *AIDS Behav.* **15**(3), 643–649 (2011)



Opinion Dynamic Modeling of Fake News Perception

Cecilia Toccaceli¹, Letizia Milli^{1,2}(✉), and Giulio Rossetti²

¹ Computer Science Department, University of Pisa, Pisa, Italy
cecitocca@gmail.com, milli@di.unipi.it

² KDD Lab. ISTI-CNR, Pisa, Italy
giulio.rossetti@isti.cnr.it

Abstract. Fake news diffusion represents one of the most pressing issues of our online society. In recent years, fake news has been analyzed from several points of view, primarily to improve our ability to separate them from the legit ones as well as identify their sources. Among such vast literature, a rarely discussed theme is likely to play uttermost importance in our understanding of such a controversial phenomenon: the analysis of fake news' perception. In this work, we approach such a problem by proposing a family of opinion dynamic models tailored to study how specific social interaction patterns concur to the acceptance, or refusal, of fake news by a population of interacting individuals. To discuss the peculiarities of the proposed models, we tested them on several synthetic network topologies, thus underlying when/how they affect the stable states reached by the performed simulations.

Keywords: Fake news · Opinion dynamics · Polarization

1 Introduction

Nowadays, one of the most pressing and challenging issues in our continuously growing and hyperconnected (online) world is identifying fake/bogus news to reduce their effect on society. Like all controversial pieces of information, fake news usually polarizes the public debate - both online and offline - with the side effect of radicalizing population opinions, thus reducing the chances of reaching a synthesis of opposing views. Moreover, such phenomena are usually amplified due to the existence of stubborn agents, individuals that foster - either for personal gain, lack of knowledge, or excessive ego - their point of view disregarding the existence of sound opposing arguments or, even, debunking evidence. So far, the leading efforts to study such a complex scenario was devoted to: (i) identifying fake news, (ii) debunk them, (iii) identifying the sources of fake news, and (iv) studying how they spread. Indeed, all such tasks are carriers of challenges as well as opportunities: each costly, step ahead increasing out knowledge on this complex phenomenon, a knowledge that can be applied to reduce its effect on the public debate. Among such tasks, the analysis of how fake news diffuse is

probably the most difficult to address. Even by restricting the analysis on the online world, tracing the path of a content shared by users of online platforms is not always feasible (at least extensively): it becomes even impossible when we consider that such content can diffuse across multiple services, of which we usually have only a partial view. However, we can argue that - in the fake news scenario - it is important how a given controversial content spreads (e.g., how different individuals get in touch with it) and how the population reached by such content perceives it. Dangerous fake news cannot only reach a broad audience, but it is also capable of convincing it of its trustworthiness. The latter component goes beyond the mere spreading process that allows it to become viral: it strictly relates to individuals' perception, opinions that are formed not only to the news content itself but also through the social context of its users.

In this work, moving from such observation, we propose a family of opinion dynamics models to understand the role of specific social factors on the acceptance/rejection of fake news. Assuming a population composed of agents aware of a given piece of information - each starting with its attitude toward it - we study how different social interaction patterns lead to the consensus or polarization of opinions. In particular, we model and discuss the effect that stubborn agents, different levels of trusts among individuals, open-mindedness and, attraction/repulsion phenomena have on the population dynamics of fake news perception.

The paper is organized as follows. In Sect. 2, the literature relevant to our work is discussed. Subsequently, in Sect. 3, we describe the opinion dynamics models we designed to describe and study the evolution of Fake news perception. In Sect. 4, we provide an analysis of the proposed models on synthetic networks having heterogeneous characteristics. Finally, Sect. 5 concludes the paper by summarizing our results and underlying future research directions.

2 Related Works

We present the literature review by dividing this Section into two subparagraphs: first, we try to characterize fake news, and we illustrate the main areas of research for these. Then, we introduce opinion dynamics, and we describe the most popular methods.

Fake News Characterization. Before examining the central studies in the literature on the topic of fake news, it is appropriate to define the term itself. There is no universal definition of fake news, but there are several explanations and taxonomies in the literature. We define “fake news” to be news articles that are intentionally and verifiably false and could mislead readers, as reported in [1]. Indeed, identifying the components that characterize fake news is an open and challenging issue [2]. Moreover, several approaches have been designed to address the problem of unreliable content online: most of them propose methods for detecting bogus contents or their creators. Focusing on the target of the analysis involving fake news, we can distinguish different areas of research:

creator analysis (e.g., bots detection [3]), content analysis (e.g., fake news identification [4]), social context analysis (e.g., the impact of the fake news and their diffusion on society [5]).

Opinion Dynamics. Recently, opinion formation processes have been attracting the curiosity of interdisciplinary experts. We hold opinions about virtually everything surrounding us, opinions influenced by several factors, e.g., the individual predisposition, the possessed information, the interaction with other subjects. In [6], opinion dynamics is defined as the process that “*attempts to describe how individuals exchange opinions, persuade each other, make decisions, and implement actions, employing diverse tools furnished by statistical physics, e.g., probability and graph theory*”. Opinion dynamics models are often devised to understand how certain assumptions on human behaviors can explain alternative scenarios, namely consensus, polarization or fragmentation. The consensus is reached when the dynamic stable state describes the population agreement toward a single and homogeneous opinion cluster; polarization describes a simultaneous presence of more than one, well defined, separated opinion clusters of suitable sizes; finally, fragmentation corresponds to a disordered state with an even higher set of small opinions’ clusters.

Agent-based modeling is often used to understand how these situations are achieved. In these models, each agent has a variable corresponding to his opinion. According to the way opinion variables are defined, models can be classified in discrete or continuous models. Among the classic models, we can distinguish: the Voter model [7], the Majority rule model [8], and the Sznajd model [9], which are discrete models that describe scenarios in which individuals have to choose between two options on a given topic (for example, yes/no, true/false, iPhone/Samsung). For the continuous models, on the other hand, the most prominent ones are the Hegselmann-Krause (HK) model [10] and Deffuant-Weisbuch model [11] that describe the contexts in which an opinion can be expressed as a real value - within a given range - that can vary smoothly from one extreme to the other, such as the political orientation of an individual.

3 Fake News: Opinion Dynamic Modeling

To model opinion dynamics of fake news perception, we assume a scenario in which a set of agents shares their position w.r.t a given piece of news (that we assume to be bogus) posted on a social platform. Agents are allowed to interact only with the contents posted by their friends, updating their point of view to account for their distance in opinions. Thus, our effort is not in estimating how the fake news spread but, conversely, in understanding how agents perceive them as a function of the social environment that surrounds them.

Due to the peculiar nature of the phenomena we are analyzing - e.g., how fake news is perceived by individuals and how such perception fosters their spreading - we opted for a continuous modeling framework, extending the well-known Hegselmann-Krause model.

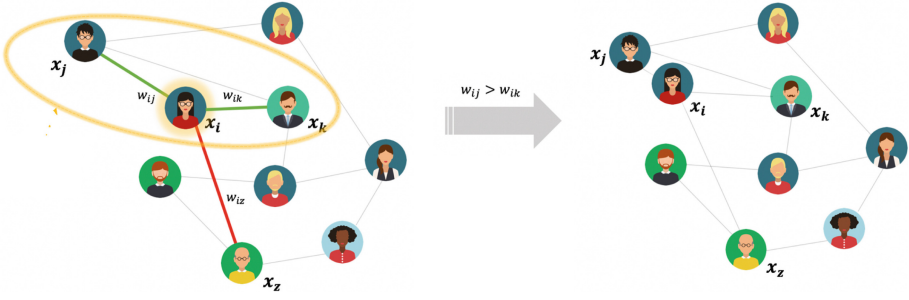


Fig. 1. *Weight example.* Opinion x_i is influenced by the opinions of agents with the opinion more similar to its opinion; e.g., the agents in the yellow elliptical. At the end of the interaction, x_i approaches the opinions of the agents with heavier weights (as visually shown x_i change of position).

Definition 1 (Hegselmann-Krause (HK)). *The HK model considers N agents - each one having an internal status representing its opinion in the continuous range $[-1, 1]$ - that interact during discrete time events, $T = \{0, 1, 2, \dots\}$. Agents can only interact if their opinions differ up to a user-specified threshold ϵ , namely their confidence level. During each interaction event $t \in T$ a random agent i is selected and the set $\Gamma_\epsilon(i)$ of its neighbors j whose opinions differ at most $d_{i,j} = |x_i(t) - x_j(t)| \leq \epsilon$ is computed. Leveraging $\Gamma_\epsilon(i)$, when selected, agent i changes its opinion following the update rule:*

$$x_i(t + 1) = \frac{\sum_{j \in \Gamma_\epsilon(i)} a_{i,j} x_j(t)}{\sum_{j \in \Gamma_\epsilon(i)} a_{i,j}} \tag{1}$$

where $a_{i,j}$ is 1 if i, j are connected by an edge, 0 otherwise. As an outcome, i 's opinion at time $t + 1$ becomes the average of its ϵ -neighbors' opinions.

The HK model converges in polynomial time, and its behavior is strictly related to the expressed confidence level: the higher the ϵ value, the higher the number of opinions clusters when model stability is reached.

Given its definition, the HK model does not consider the strength of the ties of the agents. In a fake news scenario, we can suppose that when an agent i reads a post on his Facebook wall concerning a news A the reliability attributed from i to the content of the post is closely linked to the user that shared it - as exemplified in Fig. 1. To adapt the HK model to include such specific information, we extend it to leverage weighted, pair-wise, interactions.

Definition 2 (Weighted-HK (WHK)). *Conversely from the HK model, during each iteration WHK consider a random pair-wise interaction involving agents at distance ϵ . Moreover, to account for the heterogeneity of interaction frequency*

among agent pairs, WHK leverages edge weights, thus capturing the effect of different social bonds' strength/trust as it happens in reality. To such extent, each edge $(i, j) \in E$, carries a value $w_{i,j} \in [0, 1]$. The update rule then becomes:

$$x_i(t + 1) = \begin{cases} x_i(t) + \frac{x_i(t)+x_j(t)w_{i,j}}{2}(1 - x_i(t)) & \text{if } x_i(t) \geq 0 \\ x_i(t) + \frac{x_i(t)+x_j(t)w_{i,j}}{2}(1 + x_i(t)) & \text{if } x_i(t) < 0 \end{cases} \quad (2)$$

The idea behind the WHK formulation is that the opinion of agent i at time $t + 1$, will be given by the combined effect of his previous belief and the average opinion weighed by its, selected, ϵ -neighbor, where $w_{i,j}$ accounts for i 's perceived influence/trust of j .

Moreover, we can further extend the WHK model to account for more complex interaction patterns, namely attractive-repulsive effects.

Definition 3 (Attraction WHK - (AWHK)). By "attraction", we identify those pair-wise interactions between agents that agree on a given topic. At the end of the interaction, agent i begins to doubt his position and to share some thoughts of j . For this reason his opinion will tend to approach that of his interlocutor, so $d_{ij}(t) > d_{i,j}(t + 1)$.

After selecting the pair of agents i and j , the model has the following update rule:

$$x_i(t + 1) = \begin{cases} x_i(t) - \frac{sum_{op}}{2}(1 - x_i(t)) & \text{if } x_i(t) \geq 0, x_j(t) \geq 0, x_i(t) > x_j(t) \\ x_i(t) + \frac{sum_{op}}{2}(1 - x_i(t)) & \text{if } x_i(t) \geq 0, x_j(t) \geq 0, x_i(t) < x_j(t) \\ x_i(t) + \frac{sum_{op}}{2}(1 + x_i(t)) & \text{if } x_i(t) < 0, x_j(t) < 0, x_i(t) > x_j(t) \\ x_i(t) - \frac{sum_{op}}{2}(1 + x_i(t)) & \text{if } x_i(t) < 0, x_j(t) < 0, x_i(t) < x_j(t) \\ x_i(t) - \frac{sum_{op}}{2}(1 - x_i(t)) & \text{if } x_i(t) \geq 0, x_j(t) < 0, sum_{op} > 0 \\ x_i(t) + \frac{sum_{op}}{2}(1 - x_i(t)) & \text{if } x_i(t) \geq 0, x_j(t) < 0, sum_{op} < 0 \\ x_i(t) + \frac{sum_{op}}{2}(1 + x_i(t)) & \text{if } x_i(t) < 0, x_j(t) \geq 0, sum_{op} > 0 \\ x_i(t) - \frac{sum_{op}}{2}(1 + x_i(t)) & \text{if } x_i(t) < 0, x_j(t) \geq 0, sum_{op} < 0 \end{cases} \quad (3)$$

where $sum_{op} = x_i(t) + x_j(t)w_{i,j}$.

The used criterion is always the same: the new opinion of i is the result of the combined effect of his initial opinion and that of the neighbor j , but each case applies a different formula depending on whether the opinions of i and j show discordant or not, so we can guarantee that the difference between the respective opinions is reduced after the communication.

However, when observing real phenomena, we are used to identifying more complex interactions where individuals influence each other despite their initial opinions, getting closer to the like-minded individuals and moving apart from ones having opposite views.

Definition 4 (Repulsive WHK - (RWHK)). This circumstance is called a "repulsion": two agents' opinions will tend to move them apart. Consider the

situation where agent i communicates with j with an opposite belief. At the end of the interaction, i will continue to be more convinced of his thoughts and his new opinion will be further away from that of j . So, when the communication between the two agents ends, the opinion of i will move away from that of j by following:

$$x_i(t+1) = \begin{cases} x_i(t) + \frac{sum_{op}}{2}(1 - x_i(t)) & \text{if } x_i(t) \geq 0, x_j(t) \geq 0, x_i(t) > x_j(t) \\ x_i(t) - \frac{sum_{op}}{2}(1 - x_i(t)) & \text{if } x_i(t) \geq 0, x_j(t) \geq 0, x_i(t) < x_j(t) \\ x_i(t) - \frac{sum_{op}}{2}(1 + x_i(t)) & \text{if } x_i(t) < 0, x_j(t) < 0, x_i(t) > x_j(t) \\ x_i(t) + \frac{sum_{op}}{2}(1 + x_i(t)) & \text{if } x_i(t) < 0, x_j(t) < 0, x_i(t) < x_j(t) \\ x_i(t) + \frac{sum_{op}}{2}(1 - x_i(t)) & \text{if } x_i(t) \geq 0, x_j(t) < 0, sum_{op} > 0 \\ x_i(t) - \frac{sum_{op}}{2}(1 - x_i(t)) & \text{if } x_i(t) \geq 0, x_j(t) < 0, sum_{op} < 0 \\ x_i(t) - \frac{sum_{op}}{2}(1 + x_i(t)) & \text{if } x_i(t) < 0, x_j(t) \geq 0, sum_{op} > 0 \\ x_i(t) + \frac{sum_{op}}{2}(1 + x_i(t)) & \text{if } x_i(t) < 0, x_j(t) \geq 0, sum_{op} < 0 \end{cases} \quad (4)$$

with $sum_{op} = x_i(t) + x_j(t)w_{i,j}$.

Once again, we proceed for cases, each of which defines a particular situation given by the sign of agents' opinions. The updated opinion of i will ensure that $d_{i,j}(t) < d_{i,j}(t + 1)$.

Indeed, AWHK and RWHK can be combined to obtain a comprehensive model that accounts for both behaviors.

Definition 5 (Attraction-Repulsion WHK - (ARWHK)). *To model the attraction and repulsion of opinions, during each iteration an agent i is randomly selected along with one of its neighbors, j - not taking into account the ϵ threshold. Once identified the pair-wise interaction, the absolute value of the difference between the opinions of i and j is computed. If such a value is lower than ϵ AHK is applied to compute $x_i(t + 1)$, otherwise RHK. If the difference between $x_i(t)$ and $x_j(t)$ exceeds ϵ then the repulsive interaction occurs and the update rule 4 is applied.*

The ARWHK model allows us to describe several complex scenarios and, among them, the changes of mind that individuals experience when confronted with a piece of news, either fake or not, shared by a trusted/trusted peer.

However, such a model still does not consider the existence of *stubborn* individuals - e.g., agents having fixed opinions that, despite communicating with neighboring ones, are not subject to external influence acting to influence their peers. Stubborn agents are representative of different types of individuals and are used to model those who spread misinformation.

This type of agent can correspond to prominent individuals in society, such as media, companies, or politicians. [12] and [13] are among the first studies in which the presence of this type of agent has been introduced. In the former, the system behavior is studied on homogeneous graphs for mean-field approximation; in the latter, there is an analysis based on the average of random networks and the mean-field approximation.

To integrate this idea into the model presented above, we add a binary flag to each agent to denote it as “stubborn” or not. The update rule changes are then straightforward: if the randomly selected agent is a stubborn one, he will not update his opinion and, therefore, $x_i(t) = x_i(t+1)$; otherwise, the previously discussed update strategy is applied.

4 Experimental Analysis

This Section describes the performed experimental analysis, focusing on its main components: the selected network datasets, the designed experimental protocol, and the obtained results. To foster experiments reproducibility, the introduced models have been integrated within the NDlib¹ python library [14].

Datasets. We simulate the AWHK and ARWHK models on three scenarios: (i) mean-field (e.g., complete graph), (ii) random network, and (iii) scale-free network. In all scenarios, since we are not interested in studying the proposed models’ scalability, we set the number of nodes to 100. Moreover, due to lack of space, we show the results obtained only for the networks generated with the following parameter setup: (i) Random network (Erdős and Rényi) [15], $p = 0.4$; (ii) Scale-free network (Barabasi-Albert) [16], $m = 3$.

To simulate a more realistic mesoscale network topology (e.g., presence of communities), we also tested the model against a network generated through the LFR benchmark [17]. The LFR graph is composed by 300 nodes, assigned to 4 non overlapping communities. The parameters used for its construction have been set as follows: (i) power law exponent for the degree distribution, $\gamma = 3$; (ii) power law exponent for the community size distribution, $\beta = 1.5$; (iii) fraction of intra-community edges incident to each node, $\mu = 0.1$; (iv) average degree of nodes, $\langle k \rangle = 7$; (v) minimum community size $min_s = 40$.

Analytical Protocol. The proposed model is analyzed while varying the bounded confidence, ϵ , and the percentage of stubborn agents in the network. The simulation results are then discussed through opinion evolution plots representing the evolution through each agent’s opinion.

Results. We report the results obtained by AWHK and ARWHK on the previously described synthetic scenarios and, after that, we discuss the impact of community structure on them. Edge weights, representing trust values among agent pairs, are drawn from a normal distribution.

Attraction & Stubbornness. Figure 2 shows the results obtained by AWHK on the scale-free network for different values of ϵ while maintaining constant the percentage of stubborn agents (90% of the individuals assume and maintain a positive opinion). Different colors represent the agent’s initial opinion (positive,

¹ NDlib: Network Diffusion library. <https://ndlib.readthedocs.io/>.

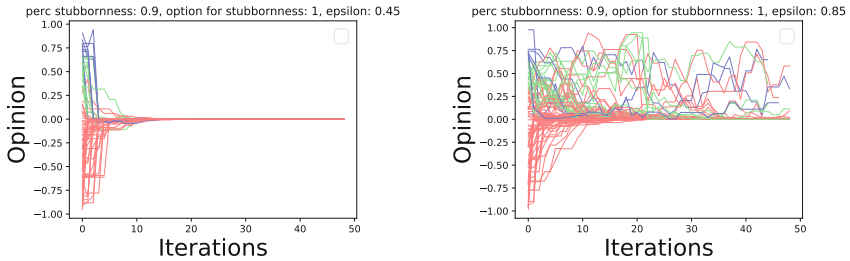


Fig. 2. Effect of the stubborn agents varying ϵ on scale-free network in the AWHK model. Stubborn population opinion evolution lines are omitted.

negative, or neutral). We can observe that in the selected scenarios, the increase of the bounded confidence interval results in a more chaotic regime, characterized by a subset of agents whose opinions heavily fluctuates toward the critical mass introduced by the stubborn agents. The presence of stubborn agents affects opinions' evolution since they act as pivots for those open to change their minds. We executed the same simulation varying the percentage of stubborns and the set of initial stubborns' opinion. As expected, we observed a similar result when stubborns are tied to negative opinions and even a more chaotic regime when such class of agents equally distributes over the opinion spectrum (we do not report the figures for limited space). So stubborns act as persuaders, bringing the opinion of the population closer to theirs. The higher their number, the more evident appears their action on the remaining population. As previously stated, Fig. 2 reports the results observed in a scale-free scenario: however, our experimental investigation underlines that the observed trends can also be identified in random and mean-field scenarios (with a significant reduction of the chaotic regime due to the more regular topological structure).

Attraction/Repulsion & Stubbornness. Figure 3 shows the simulation results obtained while introducing repulsion between the interacting subjects - as defined in the ARWHK model - while maintain constant the percentage of stubborn agents (30% of the individuals assume and maintain a negative opinion). In these settings, the overall observed while running the simulation on the scale-free network is different from what happens in the random one. In the former (highlighted in the first row of Fig. 3), we observe a fragmentation in three clusters of opinions, with the central group (the one generated by the attractive interactions), which tends to disappear by increasing the confidence parameter. In latter, when the ϵ value increases, the opinion groups tend to converge into a single one, obtaining a situation very similar to consensus. We can thus observe how the more complex scenario described by ARWHK results in more erratic behaviors. An extensive analysis of simulation results underlines that ϵ acts as a razor that implicitly separates the probability of observing either attractive or repulsive pair-wise interactions: low ϵ values will favor the application of RWHK - thus leading to a more fragmented equilibrium - while higher ones will results in

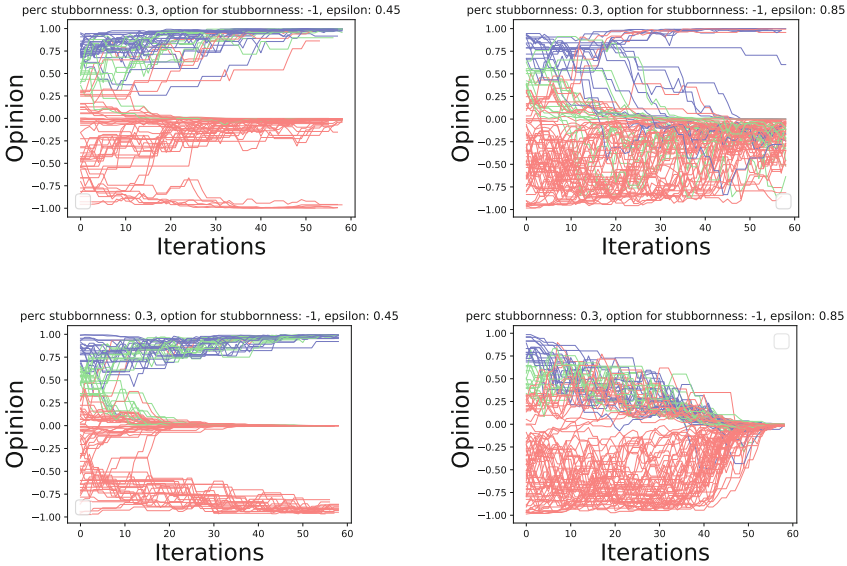


Fig. 3. Effect of the stubborn agents varying *epsilon* on scale-free (first row) and random network (second row) in the ARWHK model. Stubborn population opinion evolution lines are omitted.

a more likely application of AWHK - thus leading to consensus. However, disregarding the network topology simulating the social tissue, ARWHK convergence will require a higher number of iterations than the previously analyzed models. Moreover, even when accounting for repulsive behaviors, stubborn agents play an important role in the opinion dynamics. Our experiments suggest that their presence (i) foster the repulsive behavior for lower values of ϵ (thus increasing opinion fragmentation) and, (ii) slow-down the convergence process to a neutral opinion for higher values of such parameter.

Community Structure. To better underline node clusters' effect to the unfolding of the opinion dynamic process, we report network visualization instead of the previously adopted opinion dynamic plots. In such visualizations, nodes with positive opinions are shown in red. In contrast, the ones with negative opinions in blue: the darker the shade of colors, the more extreme opinion². In this scenario, we study the opinion spreading process while varying the number of stubborn agents and the distribution of initial opinions in the network communities. As a general remark, we observed that the stubborn agents' effect plays a relevant role only in the presence of high bounded confidence values and only when they reach high critical mass. Such a behaviour can be explained by the modular

² All images are taken from animations that reproduce the unfolding of the simulated dynamic processes. Animations, as well as the python code to generate them, are available at <https://bit.ly/3jzp1Qs>.

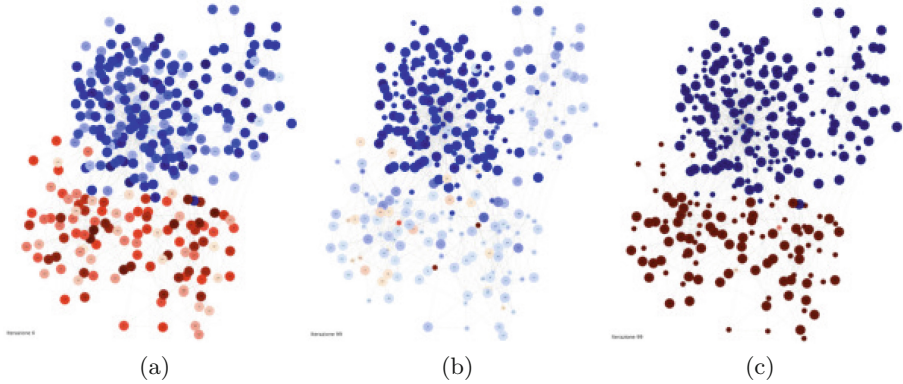


Fig. 4. Network visualizations. (a) Nodes initial conditions - three communities, two prevalently negative (blue node), two positive (red nodes); (b) AWHK final equilibrium; (c) ARWHK final equilibrium.

structure of the analyzed network that acts as boundaries for cross-cluster diffusion. The network topologies considered in this analysis are exemplified in the toy example of Fig. 4, that we will use to summarize the observed outcomes of our analysis. Such a particular case study describes a setup in which network nodes are clustered in four loosely interconnected blocks - two composed by agents sharing opinions in the negative spectrum, the others characterized by an opposite reality. In Fig. 4(a), we report the initial condition shared by two simulations (one based on AWHK, the other on ARWHK) that will be further discussed. Both simulations assume the same value for $\epsilon = 0.85$ and a fixed set of stubborn agents (e.g., the 6 less community embedded nodes - namely, the ones with the higher ratio among their intra-community degree and their total degree) - which are prevalently allocated to the bigger negative (blue) community. While performing a simulation that involves attraction, using AWHK, we can observe how the resulting final equilibrium (Fig. 4(b)) converges toward a common spectrum. In particular, in this example, we can observe how stubborn agents can make their opinion prevail, even crossing community boundaries. Indeed, such a scenario can be explained in terms of the prevalence of negative stubborn agents and the relative size of the negative communities (covering almost 3/5 of the graph). Conversely, when applying the ARWHK model, we get a completely different result, as can be observed in Fig. 4(c). Two strongly polarized communities characterize the final equilibrium. In this scenario, stubborns have a two-fold role: (i) they increase the polarization of their community by radicalizing agents' opinions and, (ii) as a consequence, make rare the eventuality of cross-community ties connecting moderate agents, thus ideologically breaking apart the population. While varying the models' parameters, our experimental analysis confirms the results obtained on the scale-free and random graphs: well-defined mesoscale clusters prevalently slow-down convergence in case of a population-wide agreement while accelerating the process of fragmentation.

5 Conclusion

In this paper, we modeled the response of individuals to fake news as an opinion dynamic process. Modeling some of the different patterns that regulate the exchange of opinions regarding a piece of given news - namely, trust, attraction/repulsion and existence of stubborn agents - we were able to drive a few interesting observations on this complex, often not properly considered, context. Our simulations underlined that: (i) differences in the topological interaction layer reflect on the time to convergence of the proposed models; (ii) the presence of stubborn agents significantly affects the final system equilibrium, especially when high confidence bounds regulates pair-wise interactions; (iii) attraction mechanisms foster convergence toward a common opinion while repulsion ones facilitate polarization.

As future work, we plan to extend the experimental analysis to real data to understand the extent to which the proposed models can replicate observed ground truths. Moreover, we plan to investigate the effect of higher-order interactions on opinion dynamics, thus measuring the effect that peer-pressure has on the evolution of individuals' perceptions.

Acknowledgment. This work is supported by the scheme 'INFRAIA-01-2018-2019: Research and Innovation action', Grant Agreement n. 871042 'SoBigData++: European Integrated Infrastructure for Social Mining and Big Data Analytics'.

References

1. Allcott, H., Gentzkow, M.: Social media and fake news in the 2016 election. *J. Econ. Perspect.* **31**(2), 211–36 (2017)
2. Zhang, X., Ghorbani, A.A.: An overview of online fake news: characterization, detection, and discussion. *Inf. Process. Manage.* **57**(2), 102025 (2020)
3. Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., Tesconi, M.: DNA-inspired online behavioral modeling and its application to spambot detection. *IEEE Intell. Syst.* **31**(5), 58–64 (2016)
4. Sharma, K., Qian, F., Jiang, H., Ruchansky, N., Zhang, M., Liu, Y.: Combating fake news: a survey on identification and mitigation techniques. *ACM Trans. Intell. Syst. Technol. (TIST)* **10**(3), 1–42 (2019)
5. Visentin, M., Pizzi, G., Pichierri, M.: Fake news, real problems for brands: the impact of content truthfulness and source credibility on consumers' behavioral intentions toward the advertised brands. *J. Interact. Market.* **45**, 99–112 (2019)
6. Si, X.-M., Li, C.: Bounded confidence opinion dynamics in virtual networks and real networks. *J. Comput.* **29**(3), 220–228 (2018)
7. Holley, R.A., Liggett, T.M.: Ergodic theorems for weakly interacting infinite systems and the voter model. *Ann. Probab.* **3**, 643–663 (1975)
8. Galam, S.: Minority opinion spreading in random geometry. *Eur. Phys. J. B-Condens. Matter Complex Syst.* **25**(4), 403–406 (2002)
9. Sznajd-Weron, K., Sznajd, J.: Opinion evolution in closed community. *Int. J. Mod. Phys. C* **11**(06), 1157–1165 (2000)
10. Hegselmann, R., Krause, U., et al.: Opinion dynamics and bounded confidence models, analysis, and simulation. *J. Artif. Soc. Soc. Simul.* **5**(3), 1–33 (2002)

11. Deffuant, G., Neau, D., Amblard, F., Weisbuch, G.: Mixing beliefs among interacting agents. *Adv. Complex Syst.* **3**(01n04), 87–98 (2000)
12. Mobilia, M.: Does a single zealot affect an infinite group of voters? *Phys. Rev. Lett.* **91**(2), 028701 (2003)
13. Wu, F., Huberman, B.A.: Social structure and opinion formation *arXiv preprint cond-mat/0407252* (2004)
14. Rossetti, G., Milli, L., Rinzivillo, S., Sirbu, A., Pedreschi, D., Giannotti, F.: NDLIB: a python library to model and analyze diffusion processes over complex networks. *Int. J. Data Sci. Anal.* **5**(1), 61–79 (2018)
15. Erdős, P., Rényi, A.: On random graphs i. *Publicationes Mathematicae Debrecen* **6**, 290 (1959)
16. Barabási, A.-L., Albert, R.: Emergence of scaling in random networks. *Science* **286**(5439), 509–512 (1999)
17. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78**(4), 046110 (2008)



Influence Maximization for Dynamic Allocation in Voter Dynamics

Zhongqi Cai^(✉), Markus Brede, and Enrico Gerding

School of Electronics and Computer Science, University of Southampton,
Southampton, UK
Zhongqi.Cai@soton.ac.uk

Abstract. In this paper, we study the competition between external controllers with fixed campaign budget in which one of the controllers attempts to maximize the share of a desired opinion in a group of agents who exchange opinions on a social network subject to voting dynamics. In contrast to allocating all the budget at the beginning of the campaign, we consider a version of a temporal influence maximization problem, where the controller has the flexibility to determine when to start control. We then explore the dependence of optimal starting times to achieve maximum vote shares at a finite time horizon on network heterogeneity. We find that, for short time horizons, maximum influence is achieved by starting relatively later on more heterogeneous networks than in less homogeneous networks, while the opposite holds for long time horizons.

Keywords: Influence maximization · Voter dynamics · Complex networks

1 Introduction

While providing new channels to guide and influence people for public benefits (e.g., health [14] or education [27]), the increasing use of social media has also led to a wider spread of fake news and misinformation [15]. Given that people's opinions can be influenced and changed by peer interactions [19] and mass media [30], it is of great importance to understand ways to guide public opinions or prevent manipulation. This problem has been formalized as the well-known topic of *influence maximization* (IM) [12]. The crux of IM is to strategically select the most influential subsets of agents in the network as the seeds to propagate a given opinion held by an external partisan (referred to as *controller* in the following) throughout the network, in order to maximize the expected number of agents adopting the opinion.

So far, a majority of research on IM are based on variants of the independent cascade (IC) model [4, 5, 10, 13]. These models simulate the propagation of influence as a one-off activation, i.e., once activated, agents keep committed to an opinion. However, in many real-world settings, individuals may repeatedly flip their opinions back and forth due to peer and media influence, e.g., attitudes

towards public or political issues. Since IC-like models only allow a single activation for each agent, they fail to address the above scenarios. Instead, *dynamic models* which allow agents to switch their opinions in both directions are suitable for modelling such an opinion formation process. In this work, we focus on the voter model because of its prominence in the literature and its conceptual simplicity in which the opinion dynamics is treated as a linear system and can be solved analytically in simple topologies (e.g., star networks) [17].

Typically, the IM problem is explored without time constraint only subject to a *budget* constraint where there are limited resources to allocate to agents in the network [4, 5, 10, 13]. However, recent research based on real-world data shows that time plays a critical role in influence propagation [7, 8]. For example, many practical applications of IM have natural time constraints. Indeed, some researchers have incorporated temporal aspects in IM [1–3, 9, 11, 16, 18]. Related to our modeling approach, Brede et al. [3] are the first to explore the IM under time constraints in voter dynamics. However, their paper does not allow controllers to allocated different amounts of resources over time, which is in contrast to real-world scenarios such as marketing [31], where the marketers can choose the start of campaigning. Representative works considering effects of time scales and activating agents depending on stages of the diffusion process include [1, 2, 11, 18]. Specifically, [2] concentrates on minimizing the diffusion time by targeting agents with different levels of connectivity at different stages of the contagion process. However, this problem is addressed in a non-competitive setting where only a single external controller spreads its influence in the network. In addition, [11, 18] explore the optimal sequential seeding for influence maximization in static and evolving networks respectively. However, both of them aim to maximize the influence in the stationary state, which is not suitable for real-world events with time limitation, and they also only investigate this problem in the presence of a single controller. Given that competition for influence is common in real-world contexts (e.g., political campaigns [29] or radicalization prevention [24]), the single-controller setting has a restricted range of applications. The only directly related study [1] solves the time-constrained IM in the presence of more than one controller by considering when to initiate opinion propagation via reinforcement learning. However, it focuses on verifying the effectiveness of the q-learning framework from an algorithmic perspective and does not explore the mechanism behind the optimal strategies. Besides, like other models discussed above, [1] is essentially static (i.e., only allow a single activation of agents) and not appropriate for modeling changing opinions.

To bridge these gaps in research about intertemporal influence allocations, we study the IM problem for *dynamic allocation* in voter dynamics under time and budget constraints in the presence of two opposing controllers. Here, we explore the dynamic allocation for the constant opponent setting where one active controller competes against a known and fixed-strategy opponent. In the context of dynamic allocations, the active controller has to design a strategy to make efficient use of its budget over time. This results in the following trade-off: If the controller starts allocating later, it has more disposable budget per unit of

time but less time left for its influence to become effective. To address this issue, we make the following contributions: (i) We are the first to define the dynamic allocation for IM in voter dynamics, where controllers have the flexibility to determine when to start control. (ii) To explore the network's influence propagation timescales, we use the heterogeneous mean-field method [3] and Taylor expansions to derive estimates for relaxation timescales towards equilibrium on scale-free networks. (iii) We demonstrate the value of our derivation and algorithm by conducting numerical experiments to address the following aspects. First, to explore the dependence of timescales towards equilibrium on network configurations, we investigate networks with different degrees of heterogeneity characterized by different degree exponents. Second, we use the interior-point optimization method [21] to obtain the optimal starting time under the time constraint.

Our main findings are as follows: (i) In constant-opponent setting, as we fix one controller to start control from the very beginning, the optimal strategy for the optimizing controller is to initially leave the system subject to the influence of the opposing controller and then only use its budget closer to the end of the campaign. (ii) For short time horizons, the optimized controller tends to start control later in highly heterogeneous networks compared to less heterogeneous networks. In contrast, for long time horizons, an earlier start is preferred for highly heterogeneous networks.

The remainder of the paper is organized as follows. In Sect. 2 we describe the model we use for dynamic allocation. In Sect. 3 we show the main results for optimal dynamic allocation. The paper concludes with a summary and future work in Sect. 4.

2 Model Description

Below, we consider social networks as graphs $G(V, E)$ where a set of N agents is identified with the vertices ($v_i \in V$) and edges $w_{ij} \in E$ indicate the strength of social connection between agent i and agent j . In line with most studies in the field, we assume an undirected and positively weighted network without self-loops. Agents in the network can hold one of two opinions: opinion A or B. Apart from the independent agents $i = 1, \dots, N$, we assume the existence of two external controllers which either favour opinion A or B, referred to as controller A and controller B. By definition, external controllers never change their opinions and aim to influence the network, such as to maximize the vote shares of their own opinions. To achieve this, subject to an overall budget constraint, both controllers can build up unidirectional connections with internal agents. In other words, the control gains $a_i(t), b_i(t)$ by controller A and controller B are time-varying unidirectional link weights which indicate the allocation of budgets by A or B to agent i at time t . As we consider the dynamic allocation of resources, the control gains $a_i(t), b_i(t)$ are functions of time and they must satisfy the budget constraints: $\sum_N \int a_i(t) dt \leq b_A$ and $\sum_N \int b_i(t) dt \leq b_B$ where b_A, b_B are the given budgets, i.e. the total amounts of resources available to the controllers.

Apart from the budget constraint, $a_i(t), b_i(t)$ also need to be non-negative, i.e., $a_i(t) \geq 0, b_i(t) \geq 0$.

To proceed, we consider the following updating process of opinions according to voting dynamics [25]. At time t , one of the agents in the network, e.g., agent i , is selected randomly. Then, agent i selects an in-neighbour or a controller at random with a probability proportional to the weight of the incoming link (including control gains from controllers). Here, we follow the mean-field rate equation for probability flows [17] by introducing x_i as the probability that agent i has opinion A. We then have:

$$\frac{dx_i}{dt} = (1 - x_i) \frac{\sum_j w_{ji} x_j + a_i(t)}{\sum_j w_{ji} + a_i(t) + b_i(t)} - x_i \frac{\sum_j (1 - x_j) w_{ji} + b_i(t)}{\sum_j w_{ji} + a_i(t) + b_i(t)}. \quad (1)$$

In this paper, we study the best strategy of controller A against a constant opponent, controller B, who starts control from the beginning of the competition. The objective function for controller A is to maximize the average vote share at time T :

$$S_A(T) = \frac{\sum_N x_i(T)}{N}. \quad (2)$$

However, due to the complexity of the non-autonomous system (Eq. (1)), it is intractable to deal with fully flexible influence allocations $a_i(t)$. In order to obtain analytical solutions for optimal allocations, we consider a simplified model where controller A only has the flexibility to determine when to start control. Once the controller starts targeting, it uniformly target all nodes. Specifically, the control gains $a_i(t), b_i(t)$ ($1 \leq i \leq N$) by controller A and controller B are:

$$a_i(t) = \begin{cases} 0 & (0 \leq t \leq t_a) \\ \frac{b_A}{(T-t_a)N} & (t_a < t \leq T) \end{cases} \quad b_i(t) = \frac{b_B}{TN} \quad (0 \leq t \leq T) \quad (3)$$

where t_a is the starting time of controller A. Consequently, this IM problem is equivalent to determining the optimal t_a that maximizes $S_A(T)$, i.e.,

$$t_a^* = \arg \max_{t_a} S_A \quad (0 \leq t_a \leq T). \quad (4)$$

Here, for numerical optimization of Eq. (4), we use the Runge-Kutta method [22] to integrate Eq. (1) and obtain the optimal starting time of controller A by interior-point optimization algorithm [21].

3 Results

We start our analysis with exploring the timescales towards equilibrium by analyzing relaxation times for networks with different degrees of heterogeneity in Sect. 3.1. Our approach is based on a heterogeneous mean-field approximation of Eq. (1). We then carry out numerical experiments to obtain the optimal strategies for the dynamic allocation in Sect. 3.2. All of our experiments are based on uncorrelated random scale-free networks with power-law degree distribution $p_k \propto k^{-\lambda}$ constructed by the configuration model [6].

3.1 Mean-Field Analysis

To obtain an analytical estimation of vote shares, we use the mean-field method [20]. In more detail, we assume that there is no assortative or dis-assortative mixing by degree. Therefore, nodes with the same degree k will have roughly similar dynamics $x_k(t)$. Here, we fix the controller B to start control at time 0. Following [3] the probability that nodes of degree k have opinion A at time t ($t > t_a$) and vote shares can be approximated as:

$$x_k(t) = \frac{a_k\alpha - \beta k + \frac{ke^{\alpha t}(\beta + \alpha x_k(t_a))}{\alpha + 1}}{\alpha(a_k + b_k + k)} - e^{-t} \left(\frac{a_k\alpha - \beta k + \frac{k(\beta + \alpha x_k(t_a))}{\alpha + 1}}{\alpha(a_k + b_k + k)} - x_k(t_a) \right) \tag{5}$$

$$S_A(t) = \sum_k p_k x_k(t) \tag{6}$$

where $\alpha = \sum_k \frac{k^2 p_k}{\langle k \rangle} \frac{1}{k + a_k + b_k} - 1$, $\beta = \sum_k \frac{k p_k a_k}{\langle k \rangle} \frac{1}{k + a_k + b_k}$, $\gamma = \sum_k \frac{k^2 p_k}{\langle k \rangle} \frac{1}{k + b_k} - 1$ and $x_k(t_a) = x_0 e^{-t_a} + \frac{k}{k + b_k} x_0 (e^{\gamma t_a} (1 - e^{-t_a}))$. Here, p_k stands for the fraction of nodes with degree k . a_k and b_k are resource allocations to nodes of degree k . Additionally, $x_k(t_a)$ is the state of nodes of degree k at time t_a . From Figs. 3 (b)(e), it can be seen that the mean-field method is a good approximation for the vote share $S_A(T)$.

According to [3], the equilibration dynamics of a node depend on its degree, which in turn, influences the optimal strategy in transient control. For example, hub nodes would typically have slow equilibration dynamics, which results in poor vote shares of hub control for short time horizons. Inspired by that, the network’s natural timescales towards equilibrium will also have an impact on determining the optimal starting time of allocations. Therefore, in the following, we investigate the dependence of equilibration dynamics in networks with different degrees of heterogeneity by systematically analyzing relaxation times as defined in [26] based on mean-field results, i.e., Eq. (5). To quantify relaxation times in the presence of various timescales, we define a normalized order parameter [26] as:

$$r_k(t) = \frac{x_k(t) - x_k(\infty)}{x(0) - x_k(\infty)}. \tag{7}$$

Here, we always use the setting that all nodes have the same initial state x_0 . Then we measure the average relaxation times for nodes with degree k via:

$$\tau_{relax,k} = \int_0^\infty r_k(t) dt = \frac{\alpha(\alpha x_0(a_k + b_k + k) - a_k\alpha + \beta k - kx_0) - \beta k}{\alpha(\alpha x_0(a_k + b_k + k) - a_k\alpha + \beta k)}. \tag{8}$$

Inserting α and β into Eq. (8) gives an involved expression. To still gain insight into the dependence of equilibration times on degree for the case that controller A and B target all nodes equally (i.e., $a_k = a$ and $b_k = b$), we approximate Eq. (8) in the limit of $\frac{a_k + b_k}{k} < 1$ up to second order and obtain

$$\tau_{relax,k} \simeq \frac{\alpha - 1}{\alpha} + \frac{a + b}{\alpha} k^{-1} - \frac{(a + b)^2}{\alpha} k^{-2} + O\left(\frac{1}{k}\right)^3. \tag{9}$$

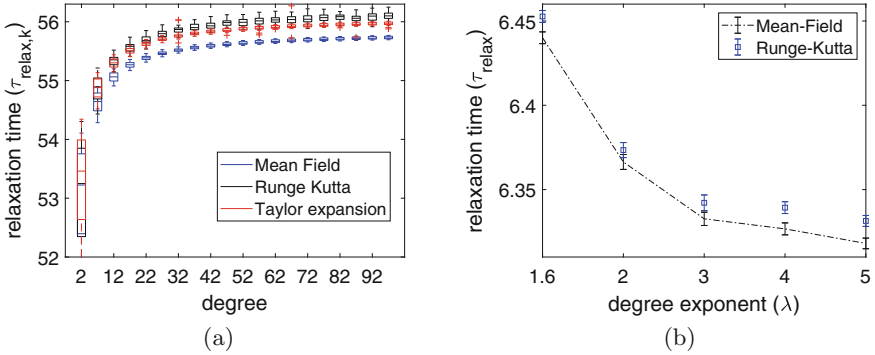


Fig. 1. Results for networks with $N = 10000$, average degree $\langle k \rangle = 10.5$, average over 100 realizations. Both controller A and controller B start control at time 0 and allocate 0.1 or 1 resource on each node per unit time respectively for Fig. 1(a) and Fig. 1(b). Figure 1(a) shows dependence of relaxation time $\tau_{relax,k}$ on degree k calculated via direct integration using a Runge-Kutta method, the mean-field estimation of Eq. (8) and a Taylor expansion of Eq. (9) in k up to 2nd order on networks with exponent 1.6. The data is represented in box plots with median, 25th and 75th percentiles and whiskers extending to the maximum or minimum values. Figure 1(b) shows dependence of relaxation time τ_{relax} on network heterogeneity calculated numerically via integration and analytically by mean-field approximation. Error bars indicate 95% confidence intervals.

The trend of $\tau_{relax,k}$ with degree k is mainly determined by the constant term $\frac{\alpha-1}{\alpha}$ and the first-order term $\frac{a+b}{\alpha}k^{-1}$. Specifically, the coefficient $\frac{a+b}{\alpha}$ is negative, which leads to the observation that the larger the degree of a node is, the longer the relaxation time will be. Moreover, the second-order term $\frac{(a+b)^2}{\alpha}k^{-2}$ reduces the difference between relaxation time for nodes of different degrees.

To proceed, we compare $\tau_{relax,k}$ calculated via direct integration using a Runge-Kutta method, the mean-field estimate of Eq. (8) and a Taylor expansion of Eq. (9) in Fig. 1 (a). From Fig. 1(a), it can be seen that x_k is monotonically increasing with degree k . This phenomenon is consistent with Gershgorin’s circle theorem [28]. According to the Gershgorin theorem, eigenvalues for nodes with degree k of Eq. (5) lie within at least the discs with radii $-1 + \frac{1}{1 + \frac{a_k + b_k}{k}}$ around zero. As we assume that the controllers target all nodes uniformly, the larger the degree of nodes, the smaller the absolute values of eigenvalues. In other words, the larger the node’s degree, the longer its relaxation time scales towards equilibrium. Additionally, we find that the mean-field method and Taylor expansion are in reasonable agreement with numerical estimates for $\tau_{relax,k}$.

Furthermore, the overall average relaxation time (i.e., network’s natural timescales towards equilibrium) for the equally targeting case is given by:

$$\tau_{relax} = \sum_k p_k \tau_{relax,k} = \sum_k p_k \frac{k(\beta(a+b) + a) + \beta(a+b)^2}{\beta(a+b)(a+b+k)} \quad (10)$$

Our next aim is to investigate the dependence of overall relaxation times on network heterogeneity characterized by the degree exponent λ . For this purpose, we numerically calculate the average relaxation time τ_{relax} for different settings of λ . Figure 1(b) shows simulation results for τ_{relax} obtained numerically via Runge-Kutta integration and compares to mean-field results based on Eq. (10). The figure illustrates that the more heterogeneous the network, the longer its timescales towards equilibrium. Combined with the results in Fig. 1 (a), we gain an intuition that the long timescales in highly heterogeneous networks are mainly caused by the higher degree nodes.

In the above, τ_{relax} only represents timescales towards equilibrium. However, we are also interested in time scales towards reaching non-equilibrium states. Therefore, we extend the notation of τ_{relax} by introducing the *degree of equilibrium* l . Here, l describes to which extent the final state approximates the equilibrium state, i.e., for $S_A(T) \leq S_A(\infty)$, $l = \frac{S_A(T)}{S_A(\infty)}$ and for $S_A(T) \geq S_A(\infty)$, $l = \frac{|2S_A(\infty) - S_A(T)|}{S_A(\infty)}$. Then, we define the average timescales towards $lS_A(\infty)$ as *l-percentage relaxation time*, given by:

$$\tau_{relax}^l = \int_0^{t'} r(t)dt = \int_0^{t'} \sum_k p_k r_k(t)dt \tag{11}$$

where t' is determined implicitly by: $S_A(t') = lS_A(\infty)$ for $x_0 \leq lS_A$ and $S_A(t') = (2 - l)S_A(\infty)$ for $x_0 \geq (2 - l)S_A(\infty)$. This equation defines an average timescale at which the vote-share dynamics approaches the desired l -percentage vote shares when the initial state x_0 is less than lS_A or greater than $(2 - l)S_A(\infty)$.

To explore the relationship between the l -percentage relaxation time and transient control, we plot the dependence of relaxation time on the degree of equilibrium l and network heterogeneity in Fig. 2 (a). We clearly see a cross-over of τ_{relax}^l in Fig. 2 (a): relaxation times are larger for less heterogeneous networks than for more heterogeneous networks for low l , but this ordering is reversed for large degree of equilibrium (see inset in Fig. 2 (a)). We hypothesize that this is a consequence of the characteristic dynamics toward equilibrium in heterogeneous networks occurring via two stages. To illustrate this point, we visualize the evolution of vote shares for high-degree and low-degree nodes in Fig. 2 (b). In more detail, we sort nodes according to their degrees in ascending order. Then we assign the first 80% as low-degree nodes and the rest as high-degree nodes according to the Pareto principle [23]. To explore which role they play in the transient dynamics, we compute the state changes $\frac{dx_i}{dt}$ grouped by low-degree nodes (i.e. $\sum_{low} \frac{dx_i}{dt}$) and high-degree nodes (i.e. $\sum_{high} \frac{dx_i}{dt}$). Then the average contribution of low-degree nodes and high-degree nodes to the vote-share changes are: $\frac{0.2 \sum_{low} \frac{dx_i}{dt}}{0.8 \sum_{high} \frac{dx_i}{dt} + 0.2 \sum_{low} \frac{dx_i}{dt}}$ and $\frac{0.8 \sum_{high} \frac{dx_i}{dt}}{0.8 \sum_{high} \frac{dx_i}{dt} + 0.2 \sum_{low} \frac{dx_i}{dt}}$. In this way, we obtain Fig. 2 (b), where we also compare vote-share changes for networks constructed for different degree exponents. In Fig. 2 (b), we see that a large proportion of vote-share changes is caused by the low-degree nodes at the beginning of the evolution. As the evolution proceeds, the dynamics of high-degree nodes

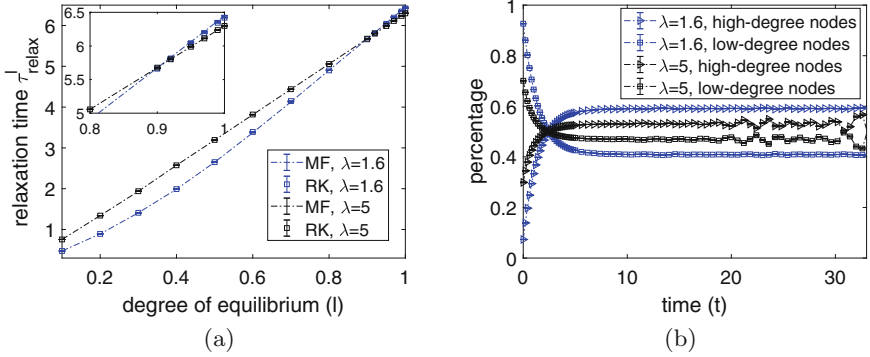


Fig. 2. Results for networks with $N = 10000$, average degree $\langle k \rangle = 10.5$, represent by error bars with 95% confidence intervals over 100 realizations. Both controller A and controller B start control at time 0 and allocate 1 resource on each node per unit time. The legend “ $\lambda = 1.6(5)$ ” is identical to power law distribution $P(k) \propto k^{-1.6(-5)}$. Figure 2 (a) shows dependence of relaxation time on degree of equilibrium l and network heterogeneity by Runge-Kutta and mean-field method. Figure 2 (b) shows evolution of average vote share changes in the proportion. “low-degree nodes” and “high-degree nodes” refer to the first 80% low degree nodes and top 20% high degree nodes. The y-axis shows the proportion of the average state changes for high-degree nodes and low-degree nodes in the total changes.

are increasingly becoming the leading cause of vote-share changes. Moreover, the degree of heterogeneity λ of the network will also make a difference in vote-share changes. For example, in the beginning, the state changes by low-degree nodes in highly heterogeneous networks make up a more significant proportion of total vote-shares changes than that by low-degree nodes in less heterogeneous networks. We thus see that the state changes by high-degree nodes in highly heterogeneous networks account for a larger proportion in total vote-shares changes than those by high-degree nodes in less heterogeneous networks.

Combining the results in Fig. 2 (a) and 2 (b), we obtain the following picture. For small l , as the state changes of vote shares are mainly driven by low-degree nodes (see the left front part of Fig. 2 (b)), the evolution of vote shares is dominated by the low-degree nodes. Since highly heterogeneous networks have much more low-degree nodes, they can approach the desired states faster. In contrast, for large l , the state changes of high-degree nodes play a crucial role in the total vote-share changes. Networks with high heterogeneity have many more high-degree nodes which thus delay the approach to equilibrium.

3.2 Optimal Strategies for Controller A

To proceed, we move on to determining optimized starting times for different time horizons T . To prove the accuracy of interior-point optimization, we present data for the dependence of vote shares on the starting time of controller A in

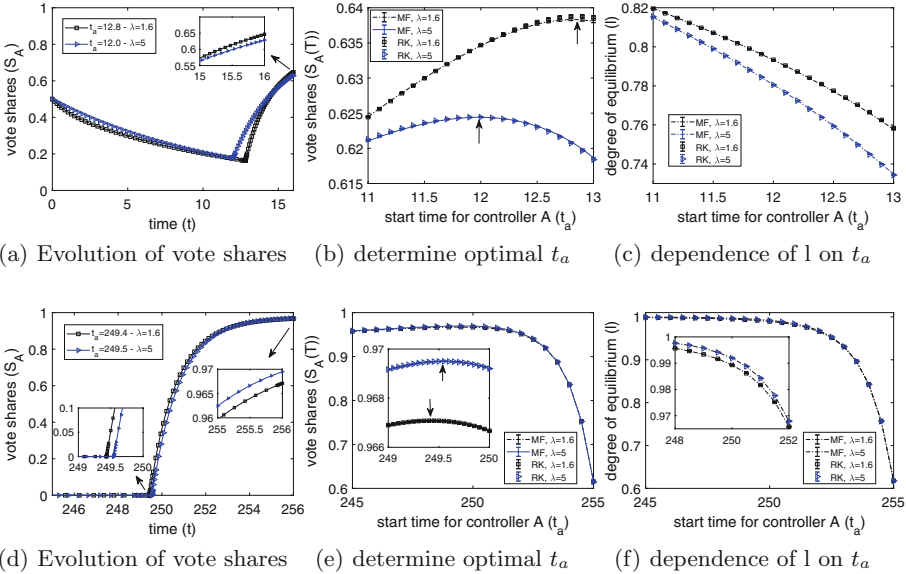


Fig. 3. Figs. (a–c) and d–f) show the evolution of total voter shares when controller A follows optimal control for time horizon $T = 16$ and $T = 256$ respectively. The turning points are the times when controller A starts control. (b) and (e) show the dependence of vote shares on controller A’s starting time for time horizons $T = 16$ and $T = 256$, respectively. (c) and (f) show the dependence of degree of equilibrium l on t_a . To find the optimal control time, the networks have to strike a balance between budget per node and degree of equilibrium. All the calculations are based on networks with $N = 10000$ and $\langle k \rangle = 10.5$ and averaged over 100 realizations. Controller B always starts its control from time 0. The black squares and blue triangles stand for networks with degree exponents $\lambda = 1.6$ and $\lambda = 5$ respectively.

Figs. 3 (b) and (e). We note that the dependence is a convex shape with a maximum, which is marked with arrows in Figs. 3 (b) and (e). The peak values of curves are consistent with the optimal starting time in Figs. 3 (a) and (d) (see the turning points in Figs. 3 (a) and (d)). In more detail, the maximum values of the vote shares is a result of a trade-off. On one hand, if t_a is small, the controller will have more time left to influence the network but with small resource allocations on each node per unit time. In other words, the final vote shares are determined by $lS_A(\infty)$. Though an early start makes the system closer to the equilibrium (i.e., l becomes larger), small resource allocations result in the small value of vote share in equilibrium (i.e., S_A becomes smaller). On the other hand, if controller A starts late, it will have more resource allocations on each node per unit time, which leads to a larger value of vote share in equilibrium, but there will be less time left for the exerted influence to become effective.

Additionally, Fig. 4 shows the dependence of the optimal starting time of the targeting controller ($T - opt\{t_a\}$) on network heterogeneity and time horizons.

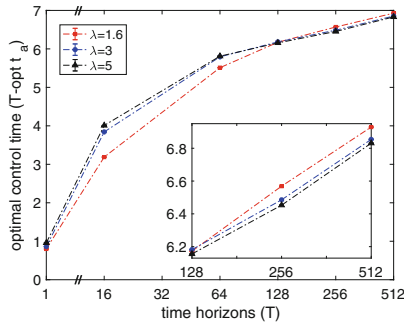


Fig. 4. Dependence of the optimal effective control time of controller A ($T - opt\{t_a\}$) on network heterogeneity and time horizons. The calculations are based on networks with $N = 10000$ and $\langle k \rangle = 10.5$ and tested in 100 realizations. The legend “1.6” is identical to $\lambda = 1.6$ for the power law distribution $P(k) \propto k^{-\lambda}$. The control gains of controller B pertaining to each nod are all fixed as 1 per unit time from time 0. The total budget of controller A are set to be the same as controller B’s, e.g., for $T = 10$, $b_A = b_B = N \times T$. The y axis shows the difference between time horizon and optimized t_a . Error bars indicate 95% confidence intervals.

Generally, the optimized controller only uses its budget near the end of the campaign. This means that the system is initially only subject to the influence of the opponent. Only when close to the end of the campaign T , the optimized controller exerts several times the allocations of its opponent on the network. In doing so, the system approaches equilibrium $\frac{a}{a+b}$ gradually, which can be seen from the monotonous rise of votes shares in Figs. 3 (a) and (d). In addition, for short time horizons, optimal control times for networks with large heterogeneity tend to start later, while for long time horizons, optimal control on highly heterogeneous networks should start slightly earlier.

This dependence of optimal starting times on network heterogeneity can be explained by our earlier observations in Fig. 2 (a). For short time horizons, the network is still far from equilibrium at the end of the competition. In other words, the network’s degree of equilibrium l is small, which corresponds to the lower-left corner of Fig. 2 (a). Therefore, the state changes of vote shares are dominated by the low-degree nodes, which have shorter timescales. As highly heterogeneous networks have more lower degree nodes, they will respond much quicker to the resource allocations. Consequently, campaigns on highly heterogeneous networks should start slightly later than on less heterogeneous networks. In contrast, for long time horizons, the network is close to equilibrium at the end of the competition. In this case, the network’s degree of equilibrium l approaches 1. From Fig. 2 (a), for a sufficiently large l , the more heterogeneous the network, the larger the relaxation time. As a result, highly heterogeneous networks respond much more slowly to resource allocations, which explains an earlier start in optimized control.

To further confirm our conclusion, we also compare the degree of equilibrium for time horizon $T = 16$ and $T = 256$ in Figs. 3 (c) and (f). To this end, Fig. 3 (c) shows that, for short time horizon $T = 16$, the degree of equilibrium is always less than 0.82. Furthermore, as shown in Fig. 2 (b), when l is less than 0.9, the relaxation time τ_{relax}^l for highly heterogeneous networks is less than that for less heterogeneous networks. In this case, the highly heterogeneous networks respond faster to the exertion of control, so control can start later. In contrast, for long time horizons, the degree of equilibrium approaches 1 (see Fig. 3 (f)). In this case, less heterogeneous networks respond faster to control. Therefore, optimal control for these networks should start later.

4 Conclusion

In this paper, we explore the IM problem under the dynamic allocation setting where controllers have the flexibility to determine when to start control. Our focus is on determining optimal starting times of campaigns on heterogeneous networks. In conclusion, our contributions are mainly threefold. (i) we extend research on transient control in the dynamic allocation setting. (ii) we analyze how the natural timescales of networks affect optimal control in networks with different degrees of heterogeneity. (iii) we numerically obtain dependence of optimal strategies on time horizons. In addition, we have obtained the following three main results. (i) the network has a natural time scale for information propagation. The controller must balance the start-up time to leave enough time for its application of control to take effect. This implies that for a network with high heterogeneity, given a short time horizon, the optimized controller will start controlling later. On the contrary, for large time horizons and highly heterogeneous networks, it is preferred to start earlier. (ii) In the constant opponent setting, by allowing the opponent to consume its budget first, the competing controllers can dominate the campaign at later stages. An interesting direction for future work is to assign different allocations as well as starting time for individual nodes under the framework of dynamic allocation.

References

1. Ali, K., Wang, C.Y., Chen, Y.S.: A novel nested q-learning method to tackle time-constrained competitive influence maximization. *IEEE Access* **7**, 6337–6352 (2018)
2. Alshamsi, A., Pinheiro, F.L., Hidalgo, C.A.: When to target hubs? Strategic diffusion in complex networks. arXiv preprint [arXiv:1705.00232](https://arxiv.org/abs/1705.00232) (2017)
3. Brede, M., Restocchi, V., Stein, S.: Effects of time horizons on influence maximization in the voter dynamics. *J. Complex Netw.* **7**(3), 445–468 (2019)
4. Budak, C., Agrawal, D., El Abbadi, A.: Limiting the spread of misinformation in social networks. In: Proceedings of the 20th International Conference on World Wide Web, pp. 665–674 (2011)
5. Carnes, T., Nagarajan, C., Wild, S.M., Van Zuylen, A.: Maximizing influence in a competitive social network: a follower’s perspective. In: Proceedings of the Ninth International Conference on Electronic Commerce, pp. 351–360 (2007)

6. Catanzaro, M., Boguná, M., Pastor-Satorras, R.: Generation of uncorrelated random scale-free networks. *Phys. Rev. E* **71**(2), 027103 (2005)
7. Gomez-Rodriguez, M., Leskovec, J., Krause, A.: Inferring networks of diffusion and influence. *ACM Trans. Knowl. Disc. Data (TKDD)* **5**(4), 1–37 (2012)
8. Goyal, A., Bonchi, F., Lakshmanan, L.V.: Learning influence probabilities in social networks. In: *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, pp. 241–250 (2010)
9. Goyal, A., Bonchi, F., Lakshmanan, L.V., Venkatasubramanian, S.: On minimizing budget and time in influence propagation over social networks. *Soc. Netw. Anal. Min.* **3**(2), 179–192 (2013)
10. Goyal, S., Heidari, H., Kearns, M.: Competitive contagion in networks. *Games Econ. Behav.* **113**, 58–79 (2019)
11. Jankowski, J., Bródka, P., Kazienko, P., Szymanski, B.K., Michalski, R., Kajdanowicz, T.: Balancing speed and coverage by sequential seeding in complex networks. *Sci. Rep.* **7**(1), 1–11 (2017)
12. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 137–146 (2003)
13. Kermani, M.A.M.A., Ardestani, S.F.F., Aliahmadi, A., Barzinpour, F.: A novel game theoretic approach for modeling competitive information diffusion in social networks with heterogeneous nodes. *Phys. A* **466**, 570–582 (2017)
14. Korda, H., Itani, Z.: Harnessing social media for health promotion and behavior change. *Health Prom. Pract.* **14**(1), 15–23 (2013)
15. Lindsay, B.R.: Social media and disasters: current uses, future options, and policy considerations (2011)
16. Liu, B., Cong, G., Xu, D., Zeng, Y.: Time constrained influence maximization in social networks. In: *2012 IEEE 12th International Conference on Data Mining*, pp. 439–448. IEEE (2012)
17. Masuda, N.: Opinion control in complex networks. *New J. Phys.* **17**(3), 033031 (2015)
18. Michalski, R., Jankowski, J., Bródka, P.: Effective influence spreading in temporal networks with sequential seeding. *IEEE Access* **8**, 151208–151218 (2020)
19. Moussaïd, M., Kämmer, J.E., Analytis, P.P., Neth, H.: Social influence and the collective dynamics of opinion formation. *PLoS ONE* **8**(11), e78433 (2013)
20. Pastor-Satorras, R., Vespignani, A.: Epidemic spreading in scale-free networks. *Phys. Rev. Lett.* **86**(14), 3200 (2001)
21. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Section 10.11. linear programming: interior-point methods. In: *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press (2007)
22. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: *Numerical recipes in c* (1988)
23. Price, D.J.D.S.: Networks of scientific papers. *Science* **30**, 510–515 (1965)
24. Ramos, M., Shao, J., Reis, S.D., Anteneodo, C., Andrade, J.S., Havlin, S., Makse, H.A.: How does public opinion become extreme? *Sci. Rep.* **5**, 10032 (2015)
25. Redner, S.: Reality-inspired voter models: a mini-review. *C.R. Phys.* **20**(4), 275–292 (2019)
26. Son, S.W., Jeong, H., Hong, H.: Relaxation of synchronization on complex networks. *Phys. Rev. E* **78**(1), 016106 (2008)
27. Tess, P.A.: The role of social media in higher education classes (real and virtual)-a literature review. *Comput. Hum. Behav.* **29**(5), A60–A68 (2013)

28. Weisstein, E.W.: Gershgorin circle theorem (2003)
29. Wilder, B., Vorobeychik, Y.: Controlling elections through social influence. In: Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems, pp. 265–273. International Foundation for Autonomous Agents and Multiagent Systems (2018)
30. Woo-Young, C.: Online civic participation, and political empowerment: online media and public opinion formation in Korea. *Media Cult. Soc.* **27**(6), 925–935 (2005)
31. Zaichkowsky, J.L., Garkey, J.R.: Defending your brand against imitation: consumer behavior, marketing strategies, and legal issues. Quorum Books Westport, CT (1995)



Effect of Interaction Mechanisms on Facebook Dynamics Using a Common Knowledge Model

Chris J. Kuhlman¹(✉), Gizem Korkmaz¹, S. S. Ravi¹,
and Fernando Vega-Redondo²

¹ Biocomplexity Institute, University of Virginia, Charlottesville, VA, USA
{cjk8gx,gkorkmaz,ssravi}@virginia.edu

² Department of Decision Sciences, Bocconi University, Milan, Italy
fernando.vega@unibocconi.it

Abstract. Web-based interactions allow agents to coordinate and to take actions (change state) jointly, i.e., to participate in collective action such as a protest, facilitating spread of contagion to large groups within networked populations. In game theoretic contexts, coordination requires that agents share common knowledge about each other. Common knowledge emerges within a group when each member knows the states and the types (preferences) of the other members, and critically, each member knows that everyone else has this information. Hence, these models of common knowledge and coordination on communication networks are fundamentally different from influence-based unilateral contagion models, such as those devised by Granovetter and Centola. Common knowledge arises in many settings in practice, yet there are few operational models that can be used to compute contagion dynamics. Moreover, these models utilize different mechanisms for driving contagion. We evaluate the three mechanisms of a common knowledge model that can represent web-based communication among groups of people on Facebook. We evaluate these mechanisms on five social (media) networks with wide-ranging properties. We demonstrate that different mechanisms can produce widely varying behaviors in terms of the extent of contagion spreading and the speed of contagion transmission.

Keywords: Common knowledge · Coordination · Social networks · Contagion models · Facebook

1 Introduction

1.1 Background and Motivation

Infamous waves of uprisings (e.g., Black Lives Matter, Women’s March, Occupy Wall Street) are commonly characterized by the significant use of social media to share information prior to, as well as during, protests to reach a critical

number of participants. The goal of understanding how local online interactions through social networks can facilitate information sharing in a way that generates common knowledge and coordination within large groups has motivated the construction of models of mobilization. While the exemplar in this work is protests, other applications of mobilization are family decisions to evacuate in the face of hurricanes and forest fires, and to participate in demonstrations for equality. The results herein apply to these examples as well.

There are many influence-based threshold models of diffusion that have been proposed and evaluated, e.g., [8, 9, 11, 17, 21]. In a networked population, an agent or network node i transitions from an inactive state (state 0) to an active state (state 1) if at least a threshold θ number of its neighbors (connections) are already in state 1. These models are used to explain different behaviors, such as the spread protests [8] and Twitter hashtags [17]. Watts argues for the use of threshold models in a wide range of scenarios [21]. In these models, agents make individual decisions to change state, irrespective of the decisions of their neighbors, and hence are referred to as *unilateral* models.

In contrast, in game-theoretic models of collective action, agents' decisions to transition to state 1 depend on their expectations of what others will do. That is, they need to know each others' willingness to participate (defined by the threshold θ) and this information needs to be *common knowledge* among a group of agents. Common knowledge (CK) emerges within a group when each member knows the states and attributes (e.g., preference, type) of the other members, and critically, each member knows that everyone else knows her attributes. Common knowledge enables a group of agents to *coordinate* their actions, thus enabling them to transition state simultaneously if it is mutually beneficial to do so.

In the context of collective action, e.g., protests, two CK models ([5] and [13]) combine social structure and individual incentives together in a coordination game of incomplete information and provide a rigorous formalization of common knowledge. The authors study which network structures are conducive to coordination, and the local spread of knowledge and collective action.

CK models are fundamentally different from unilateral models as *(i)* contagion can *initiate* in CK models—meaning that contagion can be generated when no contagion previously existed—whereas it does not in unilateral models (unless an agent's threshold is zero); *(ii)* CK models may utilize multiple mechanisms at graph geodesic distances of 1 and 2, whereas unilateral models most often use influence from distance-1 neighbors, and *(iii)* the characterizing (social) network substructure for threshold-based models is a star subgraph centered at the ego node making a decision, while those for CK models include distance-2 based stars and other substructures such as cliques [6] and bicliques [13] (i.e., complete bipartite graphs).

In this work, we evaluate the Common Knowledge on Facebook (CKF) model [13]. It models communication on Facebook (through “wall” or “timeline”) as a means to generate CK and to facilitate coordination. Geodesic distance-2 communication is achieved as follows: two individuals i and j do not directly communicate, but each communicates with person k . This means that if, for

Table 1. Communication mechanisms of the CKF model evaluated in this work, individually and in combination. These mechanisms may be operative in contagion initiation, propagation, or both. Mechanism abbreviations are denoted in [·].

Mechanism	Description
Common knowledge [CK]	This is a common knowledge mechanism characterized by <i>bicliques</i> in social networks. This mechanism can <i>initiate</i> contagion, and can drive contagion <i>propagation</i> . No seeded nodes with contagion are required.
Neighborhood dynamics [ND2]	This is influence (communication) produced by neighbors within distance-2 of an ego node. This mechanism <i>propagates</i> contagion.
Population dynamics [PD2]	Since agents (nodes) know both states and thresholds of agents within distance-2, an agent can infer information about the numbers of agents currently in state 1, even when these other agents are at geodesic distances of 4 or more. This mechanism <i>propagates</i> contagion.

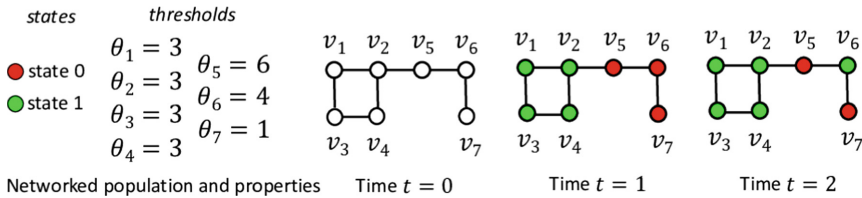


Fig. 1. Spread of contagion on a 7-node graph illustrating the mechanisms of Table 1. Each operative mechanism is evaluated independently, at each t . At $t = 2$, the *spread size* is 5 (5 nodes in green), and the *spread fraction* is $5/7$. The dynamics resulting from the different mechanisms are discussed in the text.

example, i writes information about herself on k 's wall, then j knows i 's information by reading k 's wall, without directly communicating with i . The information thus travels distance-2, from i to k to j . Multiple mechanisms are operative in the CKF model, including CK itself, network dynamics, and local and global interactions. Hence, it is of interest to understand the effects of mechanisms on the spread of contagions. We aim to develop computational models of the CKF model mechanisms to study these mechanisms individually and in combination, to quantify their effects on the spread of collective action. Table 1 describes these mechanisms, which are formalized in Sect. 3.

Figure 1 provides an example illustrating all three mechanisms summarized in Table 1. In this network, there are 7 people with different thresholds. Based on

the CKF model [13] summarized in Sect. 3, for agents to participate (i.e., transition to state 1), they need to share common knowledge with a group of people (they need to form a complete bipartite graph), and their thresholds should be less than the size of the common knowledge set (i.e., the group they share common knowledge with). In this example, agents 1, 2, 3, and 4 have threshold of 3, indicating that each needs to have at least 3 other people to participate (i.e., transition to state 1) for them to participate. These four people form a complete bipartite graph (a square) that allows them to generate common knowledge about their willingness to participate. They know each others' thresholds and know that they are sufficiently low for them to jointly participate and achieve mutual benefits. Hence, they transition to state 1 at $t = 1$. This is referred to as the *common knowledge* [CK] mechanism. On the other hand, agent 5, who shares common knowledge of thresholds with agents 1, 2, and 4 (through the 4-node star network centered at agent 2), has threshold of 6 which is not low enough for him to participate with the other 3 players that he shares CK with. Agent 5 also is part of CK node sets $\{2, 5, 6\}$ (a 3-node star centered at agent 5) and $\{5, 6, 7\}$ (a 3-node star centered at agent 6), but cannot transition to state 1 for the same reason. Similarly, persons 6 and 7 do not transition to state 1 at $t = 1$.

Since agent 2 is within distance-2 of agent 6 (friend-of-friend), agent 6 knows agent 2's threshold and state (action), through the Facebook wall or timeline of agent 5. At $t = 2$, agent 2's state is 1 and her fixed threshold is 3. Thus, agent 6 knows that at least four agents are in state 1. Agent 6's threshold is satisfied and she transitions to state 1. This is the *population dynamics* [PD2] mechanism.

Finally, at $t = 3$, person 7 will transition to state 1 as a result of the *neighborhood dynamics* [ND2] mechanism: it has one activated neighbor (agent 6) within distance-2 to meet its threshold of 1. All of the state transitions in this example are made formal in Sect. 3.

1.2 Contributions of This Work

Following others who study contagion dynamics on networks (e.g., [19]), we quantify contagion dynamics on five web-based social networks that range over $6\times$ (i.e., over a factor of 6) in numbers of nodes, $4\times$ in numbers of edges, $4\times$ in average degree, $13\times$ in maximum degree, and $80\times$ in average clustering coefficient. Thresholds range over $3\times$. We construct agent-based models and a framework that can turn on and off any combination of mechanisms in simulations of contagion dynamics. (The CKF model is presented in Sect. 3; simulation process is given in Sect. 5.) There are also companion theoretical results, but owing to space limitations, these will be included in an extended version of the paper.

1. Effects of different contagion mechanisms on the spread evolution.

We demonstrate that: (i) The [ND2] mechanism, as a driving force for contagion diffusion, is often relatively weak compared to the other mechanisms. For many networks and sets of simulation parameters, plots of fractions of nodes in state 1,

as a function of time, show little difference between the effects of the [CK] mechanism alone, versus the [CK] and [ND2] mechanisms combined. However, there are cases (e.g., for the P2P network with $\theta = 8$ and $p_p = 0.2$), where the addition of [ND2] to [CK] increases the spread fraction by more than 50%. (ii) The [PD2] mechanism dominates the other two mechanisms for driving contagion in particular cases (e.g., for the Facebook (FB) network). In other cases, the [CK] mechanism dominates (e.g., for several cases for the Wiki network). As average degree decreases relative to threshold, the more the [PD2] mechanism can dominate. As average network degree increases, the more the [CK] mechanism dominates. This is because a star subgraph is a form of biclique, and the more nodes in a biclique, the more threshold assignments will cause state transition owing to CK. (iii) If [CK] and [PD2] mechanisms are already operative, then there is no increase in spread fraction if mechanism [ND2] is incorporated. (iv) There are combinations of simulation conditions (e.g., network, threshold, participation probability, CK model mechanisms) that can produce small or large spread size changes by varying only one of these inputs.

2. Sensitivity of contagion dynamics to average degree d_{ave} . The spread size (large or small) is driven by the magnitude of average degree relative to the node threshold θ assigned uniformly to all nodes. In all five networks, spread size can be large (e.g., spread fraction > 0.5). If $d_{ave} > \theta$, then outbreak sizes are large; if $d_{ave} < \theta$, then outbreak sizes are lesser. We demonstrate a pronounced effect on spread size even when the magnitudes of d_{ave} and θ are close.

2 Related Work

There are several studies that model web-based social media interactions, including the following. The spread of hashtags on Twitter is modeled using a threshold model in [17]. Diffusion on Facebook is modeled in [18], and a similar type of mechanism on Facebook is used to study the resharing of photographs [4]. None of these works uses the “wall” or “timeline” mechanism of Facebook that is modeled here in the CKF model. Several *unilateral* models and applications were identified in the Introduction. These are not repeated. Here, we focus on game-theoretic common knowledge models, in particular, the CKF model.

A couple of data mining studies have used Facebook walls, including an experimental study [7]. Features of cascades on Facebook are studied using user wall posts [10], but again, these are cascades of the conventional social influence type; there is no assessment of CK-based coordination.

There have been a few works on the CKF model, which was initially introduced in [13]. Details of the game-theoretic formulation are provided there. For example, the CKF model is not efficiently computable because finding all bicliques in a network is an NP-hard problem. This makes studying CKF on very large networks (e.g., with 1 million or more nodes) extremely difficult. An approximate and computationally efficient CKF model is specified in [14]. CK dynamics on networks that are devoid of key players is studied in [15]. *None* of these investigates the individual and combinations of mechanisms of Table 1.

3 Model

3.1 Preliminaries

This section provides a formal description of the Common Knowledge on Facebook (CKF) model [13] studied in this paper. The population is represented by a communication network $G(V, E)$. There is a node set $V = \{1, 2, \dots, n\}$ of n nodes (people) and edge set E where an undirected edge $\{i, j\} \in E$ means that nodes $i, j \in V$ can communicate with each other. Each person $i \in V$ is in a state $a_{it} \in \{0, 1\}$ at time t : if $a_{it} = 1$, person i is in the active state (e.g., joining a protest), and $a_{it} = 0$ otherwise (e.g., staying at home). We use progressive dynamics [11], such that once in state 1, nodes do not transition back to 0. Each node i has a threshold θ_i that indicates its inclination/resistance to activate. Given person i 's threshold θ_i and the system state at t , denoted by $a_t = (a_{1t}, a_{2t}, \dots, a_{nt})$, her utility is given by

$$U_{it} = \begin{cases} 0 & \text{if } a_{it} = 0 \\ 1 & \text{if } a_{it} = 1 \wedge \#\{j \in V : a_{jt} = 1\} \geq \theta_i \\ -z & \text{if } a_{it} = 1 \wedge \#\{j \in V : a_{jt} = 1\} < \theta_i \end{cases} \quad (1)$$

where $-z < 0$ is the penalty she gets if she activates and not enough people join her. Thus, a person will activate as long as she is sure that there is a sufficient number of people (in the population) in state 1 at t . A person always gets utility 0 by staying in state 0 regardless of what others do since we do not consider free-riding problems. When she transitions to the active state, she gets utility 1 if the total number of other people activating is at least θ_i . (Note that these "others" do not have to be neighbors of i , as in unilateral models.)

The CKF model describes Facebook-type (friend-of-friend) communication in which friends write to and read from each others' Facebook walls and this information is also available to their friends of friends. The mechanisms and its implications are described below. The communication network indicates that if $\{i, j\} \in E$, then node i (resp., j) communicates (θ_i, a_{it}) (resp., (θ_j, a_{jt})) to node j (resp., i) over edge $\{i, j\}$ at time t , and this information is available to j 's (resp., i 's) neighbors. The communication network helps agents to coordinate by creating common knowledge at each t . Agents' presence on the network (online or offline) is captured by the participation probability $0 \leq p_p \leq 1$ for each node, which determines whether a node is participating in the contagion dynamics at each t ; e.g., whether i is online or offline at t in Facebook.

3.2 Facebook Common Knowledge Model Mechanisms

Here we describe the three mechanisms in this model (cf. Table 1), and their implications. Figure 1 illustrates these mechanisms through an example. First of all, the CKF model describes a Facebook type communication which allows for

distance-2 communication: two nodes, i and j , with $\{i, j\} \notin E$ can communicate by posting to and reading from the wall of a common neighbor k , provided $\{i, k\}, \{j, k\} \in E$. Thus, all $i \in V$ can communicate with all nodes $j \in V$ such that their geodesic distance is $|\{i, j\}| \leq 2$. All three mechanisms make use of this Facebook communication structure.

The **neighborhood dynamics [ND2]** mechanism (Table 1) is similar to the Granovetter [9] unilateral contagion model, but with interaction at distance-1 and -2. Let the neighbors j of i within distance-2 be defined by $N_i^2 = \{j : |\{i, j\}| \leq 2\}$. The [ND2] mechanism is given by

$$a_{it} = \begin{cases} 1 & \text{if } a_{i,t-1} = 1 \text{ or } |\{j \in N_i^2 : a_{j,t-1} = 1\}| \geq \theta_i \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

For the **common knowledge [CK]** mechanism of Table 1, the biclique subgraph is the structure necessary for creation of CK among a group of people [13], and allows them to jointly activate. We first compute all node-maximal bicliques in G , which is an NP-hard problem [2]. Let $M^{biclique}$ denote a set of nodes of G that forms a biclique. Then, V in Eq. (1) is replaced with $M^{biclique}$. At each t , Eq. (1) is computed for each $i \in V$ in each CK set $M^{biclique}$ for which $i \in M^{biclique}$.

Finally, the **population dynamics [PD2]** mechanism indicates that a node i that is in state 0 can infer a minimum number of nodes already in state 1 if a neighbor j in N_i^2 is already in state 1, by knowing θ_j . Formally,

$$a_{it} = \begin{cases} 1 & \text{if } a_{i,t-1} = 1 \text{ or} \\ & (\max \theta_j : j \in N_i^2, a_{j,t-1} = 1) + 1 \geq \theta_i \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Assume $a_{i,t-1} = 0$. If $j \in N_i^2$ and $a_{j,t-1} = 1$, with θ_j , then i can infer that at least $\theta_j + 1$ nodes are in state 1. Now, if $\theta_i \leq \theta_j + 1$, then i will transition to state 1; i.e., $a_{it} = 1$.

At each time $t - 1$, all operative mechanisms are evaluated, independently, for each $i \in V$ for which $a_{i,t-1} = 0$. If any of the three mechanisms causes i to transition, then $a_{it} = 1$.

4 Social Networks

The web-based networks of this study are summarized in Table 2. FB is a Facebook user network [20], P2PG is a peer-to-peer network, Wiki is a Wikipedia network of voting for administrators, and Enron is an Enron email network [16]. All but the SF1 network are real (i.e., mined) networks. SF1 is a scale free (SF) network generated by a standard preferential attachment method [3] to fill in gaps of the real networks. For networks possessing multiple components,

we use the giant component. These networks have wide-ranging properties and hence represent a broad sampling of web-based mined network features. Figure 2 shows the average degrees per network in the original graphs G , corresponding to geodesic distance of 1, and in the square of the graphs G^2 that are particularly relevant to CK model dynamics (forthcoming in Sect. 6).

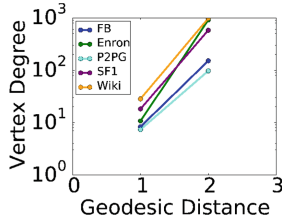


Fig. 2. Average vertex degree for geodesic distances 1 and 2 (i.e., for G^1 and G^2), which are relevant for the CK, ND2, and PD2 mechanisms for driving contagion through networks.

5 Agent-Based Model and Simulation Parameters

We conduct discrete time agent-based simulations based on the model described in Sect. 3 using the web-based networks given in Table 2. Table 3 summarizes the parameters and their values associated with each simulation. A **simulation** consists of a set of 30 runs, where a **run** consists of the spread of contagion from an initial configuration (or state) with all nodes in state 0 at time $t = 0$, to a specified maximum time t_{max} . Differences among runs is stochasticity in models.

Table 2. Characteristics of web-based social networks analyzed. If there are multiple connected components in a graph, we use only the giant component. Here, n and m are numbers of nodes and edges, respectively; d_{ave} and d_{max} are average and maximum degrees; c_{ave} is average clustering coefficient; and Δ is graph diameter. Properties are computed with the codes in [1].

Network	Type	n	m	d_{ave}	d_{max}	c_{ave}	Δ
FB	Facebook	43,953	182,384	8.30	223	0.115	18
P2P	Peer Comms.	10,876	39,940	7.34	103	0.00622	10
Enron	Email	33,696	180,811	10.7	1,383	0.509	17
SF1	Stylized	4,956	45,031	18.2	270	0.0780	8
Wiki	Online Voting	7,115	100762	28.3	1065	0.141	7

Table 3. Summary of contagion study parameters.

Parameter	Description
Agent Thresholds θ	Uniform threshold values for a simulation: all nodes in a network have the same value. Values range from $\theta = 8$ through $\theta = 29$.
Participation Probabilities p_p	Uniform value for all nodes in a simulation. Values in the range of 0.05 to 0.4.
Model Mechanisms	[CK], [ND2], and [PD2] mechanisms described in Table 1. [CK] is always operative to initiate contagion.
Seed Vertices	No specified seed vertices; all vertices initially in state 0. CK model initiates contagion without seeds.
Simulation Duration t_{max}	30 and 90 time steps.

6 Simulation Results

In this section, we present the results of our agent-based model simulations. All results provided are average results from 30 runs.

Effects of CKF model mechanisms on contagion dynamics. We analyze the effects of the [CK], [ND2], and [PD2] mechanisms (described in Table 1) and their combinations on the time histories of activated nodes for each network of the study. Figure 3 contains time histories for the fraction of nodes in state 1 over time for the Wiki network. In this simulation, all nodes have threshold $\theta \approx d_{ave} = 29$. The mechanism combinations are [CK] only, [CK] plus [ND2], [CK] plus [PD2], and [CK] plus [ND2] plus [PD2] (i.e., all) mechanisms. In Fig. 3a, $p_p = 0.1$; in Fig. 3b, $p_p = 0.4$. Several observations are important. First, the [ND2] mechanism does not contribute significantly to the driving force to transmit contagion in the system. This is seen in the first plot in that the magenta curve is only slightly above the blue curve, i.e., the addition of [ND2] to [CK] results in a small increase in spread fraction (i.e., fraction of agents in state 1). In comparing the orange and green curves in the left plot, we observe that adding [ND2] to [CK]+[PD2] does not increase spread fraction. The same two comparisons in the right plot (where $p_p = 0.4$) give the same conclusion. Second, the addition of the [PD2] mechanism to the [CK] mechanism can produce significant increases in spread fraction (comparing blue and green curves). Third, in moving from the left to the right plot, the spread fractions increase, for a given time t , and the contagion spreads more rapidly, with increasing p_p . These findings are shown for all networks, as described below.

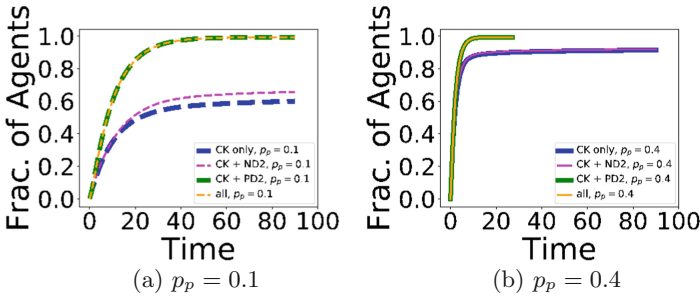


Fig. 3. Wiki network results for $\theta = d_{ave} = 29$: (a) $p_p = 0.1$, (b) $p_p = 0.4$. Cumulative fraction of agents in state 1 is plotted as a function of time in simulation for combinations of different mechanisms (in Table 1). Each propagation mechanism is isolated for different simulations and is represented by a different curve; however, [CK] (labeled CK) is always operative. In (a), the blue ([CK] only) and magenta ([CK + [ND2]]) curves are close together, indicating that for $p_p = 0.1$, the distance-2 classic diffusion mechanism [ND2] provides a relatively small increase to the overall contagion driving force. In (b), the blue ([CK] only) and magenta ([CK + [ND2]]) curves overlay; this means that [ND2] provides no noticeable increase in driving force for contagion spreading. [PD2] (green and orange curves) in both plots provides significant additional driving force, since the green and orange curves are well above the blue and magenta curves. The *all mechanisms* (denoted *all* in legend) curves coincide with the [CK]+[PD2] curves since they (orange curves) overlay with the green curves. This means that, again, [ND2] does not provide much driving force to spread a contagion.

The effect of CK-only mechanism on contagion dynamics compared to the full model across networks. We analyze the fraction of activated nodes over time under the CK-only mechanism and under all mechanisms of the CKF model combined. Figure 4 provides results for all networks, where the agent threshold in all networks is $\theta = 9$. In Fig. 4a, the networks with greater outbreaks (Enron, Wiki, and SF1) have average degrees greater than θ , while those with lesser outbreaks (FB and P2P) have values of d_{ave} that are lesser than θ . It is worth noting that three networks have d_{ave} near θ . In Fig. 4b, the addition of [ND2] and [PD2] driving forces results in a relatively greater increase in the spread size for P2P. Since FB has greater d_{ave} than P2P, one would surmise that FB should also show increased spreading in Fig. 4b. The reason this is not the case is that P2P has far more nodes with degree 10 than does FB, and thus the driving force for $\theta = 9$ is greater in the P2P network.

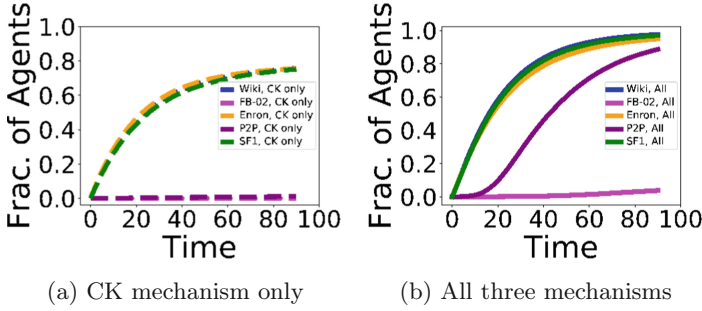


Fig. 4. Cumulative fraction of agents in state 1 as a function of simulation time, for $p_p = 0.05$ and for $\theta = d_{ave} = 9$: (a) only [CK] is active, (b) all mechanisms are active. See Table 1. The results show the sensitivity of outbreak size on average degree d_{ave} . In (a), d_{ave} for FB and P2P are slightly less than θ ; these networks have small outbreaks due to CK only. SF1, Enron, and Wiki all have d_{ave} s greater than $\theta = 9$, one (Enron) only slightly, and the other two have d_{ave} appreciably greater than θ . In (b), the addition of [PD2] drives the contagion to greater magnitudes.

Comparisons of final contagion spread at time $t = 30$ across networks.

Finally, Fig. 5 provides spread fractions at $t = 30$ for four of the five networks under different combinations of mechanisms (specified on x-axis): from left to right, [CK] only, [CK]+[ND2], [CK]+[PD2], and all three mechanisms combined. The uniform threshold for each network is its average degree, so that $\theta = d_{ave}$ is different across networks. In each plot, curves are for $p_p = 0.05, 0.1, 0.2, 0.4$.

The FB network of Fig. 5a has the smallest spread sizes. The [CK] mechanism in isolation can drive contagion through appreciable fractions of the other three networks, depending on p_p . FB, and Enron in Fig. 5b, show no effect of the [ND2] mechanism on spread fractions. However, P2P and Wiki in Figs. 5c and 5d show positive contributions to spread size from the [ND2] mechanism. It is remarkable for P2P (Fig. 5c) when $p_p = 0.2$. In all four plots, the [PD2] mechanism contributes significantly to the driving force for contagion spread (the positive slopes of curves from “+ND2” to “+PD2” on the x-axis), except perhaps when [CK] or [CK]+[ND2] produce very large spread sizes. Finally, we observe that the curves are flat in going from “+PD2” to “All” on the x-axis, where the difference is the addition of the [ND2] mechanism.

There is intuition for the lesser effectiveness of the [ND2] mechanism, relative to [PD2]. When p_p is low, a vertex in state 0 can have relatively fewer neighbors within distance-2 that are participating. The [ND2] mechanism counts the number of these neighbors that are in state 1, and hence the mechanism is weaker. In contrast, for [PD2], a node i in state 0 needs only *one* participating and active neighbor j within distance-2 that has a threshold $\theta_j + 1 \geq \theta_i$ in order for i to change state to 1. This is a stronger mechanism, and hence the spread is greater.

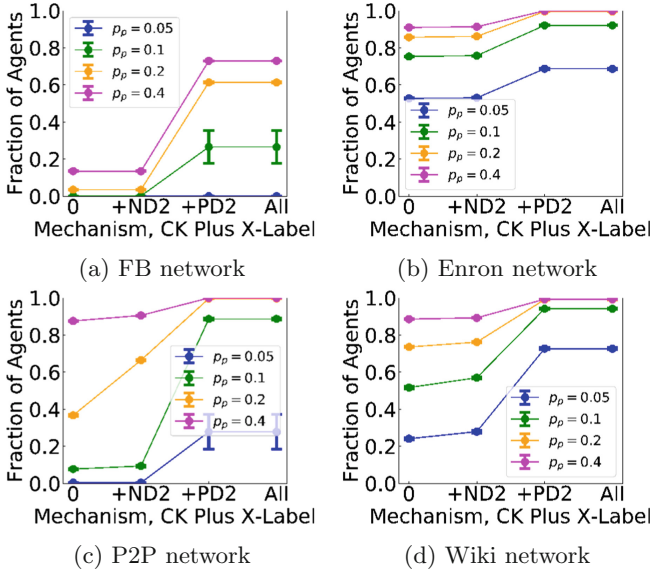


Fig. 5. CKF model results. Cumulative fraction of agents in state 1 at time $t = 30$ as a function of mechanisms and p_p (same legend for all plots) for $\theta = d_{ave}$: (a) FB, $\theta = 9$; (b) Enron, $\theta = 11$; (c) P2P, $\theta = 8$; and (d) Wiki, $\theta = 29$. The mechanisms on the x-axis always includes [CK] over all 30 time steps, where “0” corresponds to only the [CK] mechanism; “+ND2” means [CK] and [ND2]; “+PD2” means [CK] and [PD2]; and “All” means the full model. The error bars for y-axis values represent one stdev. The data illustrate that [PD2] provides a much greater driving force for contagion spread than does [ND2]. Although [CK] initiates contagion, [PD2] often generates a greater contribution to driving force than does [CK]. See for example P2P and $p_p = 0.1$.

7 Conclusion

We evaluate the CKF contagion model on a set of networks with wide ranging properties, for a range of thresholds and participation probabilities. We model and investigate multiple mechanisms of contagion spread (initiation and propagation), as well as the full model. We find evidence that the [CK] and [PD2] mechanisms are the major driving forces for the contagion initiation and spread, compared to [ND2]. These types of results are being used to specify conditions for impending human subject experiments that will evaluate CK and its mechanisms (e.g., [12]), and will be used to assess the predictive ability of the models.

Acknowledgments. We thank the anonymous reviewers for their helpful comments. This material is based upon work supported by the National Science Foundation (NSF IIS-1908530, NSF OAC-1916805, and NSF CRISP 2.0 Grant 1832587) and the Air Force Office of Scientific Research under award number FA9550-17-1-0378. Any opinions, finding, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

References

1. Ahmed, N.K., Alo, R.A., Amelink, C.T., et al.: net.science: a cyberinfrastructure for sustained innovation in network science and engineering. In: Gateway (2020)
2. Alexe, G., Alexe, S., Crama, Y., Foldes, S., Hammer, P.L., Simeone, B.: Consensus algorithms for the generation of all maximal bicliques. *DAM* **145**, 11–21 (2004)
3. Barabasi, A., Albert, R.: Emergence of scaling in random networks. *Nature* **286**, 509–512 (1999)
4. Cheng, J., Adamic, L.A., Dow, P.A., Kleinberg, J., Leskovec, J.: Can cascades be predicted? In: WWW (2014)
5. Chwe, M.S.Y.: Communication and coordination in social networks. *Rev. Econ. Stud.* **67**, 1–16 (2000)
6. Chwe, M.S.Y.: Structure and strategy in collective action. *Am. J. Sociol.* **105**, 128–156 (1999)
7. Devineni, P., Koutra, D., Faloutsos, M., Faloutsos, C.: If walls could talk: Patterns and anomalies in facebook wallposts. In: ASONAM, pp. 367–374 (2015)
8. Gonzalez-Bailon, S., Borge-Holthoefer, J., Rivero, A., Moreno, Y.: The dynamics of protest recruitment through an online network. *Sci. Rep.* **1**, 1–7 (2011)
9. Granovetter, M.: Threshold models of collective behavior. *Am. J. Soc.* **83**(6), 1420–1443 (1978)
10. Huang, T.K., Rahman, M.S., Madhyastha, H.V., Faloutsos, M., et al.: An analysis of socware cascades in online social networks. In: WWW, pp. 619–630 (2013)
11. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: KDD, pp. 137–146 (2003)
12. Korkmaz, G., Capra, M., Kraig, A., Lakkaraju, K., Kuhlman, C.J., Vega-Redondo, F.: Coordination and common knowledge on communication networks. In: Proceedings of the AAMAS Conference, 10–15 July, pp. 1062–1070, Stockholm (2018)
13. Korkmaz, G., Kuhlman, C.J., Marathe, A., et al.: Collective action through common knowledge using a Facebook model. In: AAMAS (2014)
14. Korkmaz, G., Kuhlman, C.J., Ravi, S.S., Vega-Redondo, F.: Approximate contagion model of common knowledge on Facebook. In: Hypertext, pp. 231–236 (2016)
15. Korkmaz, G., Kuhlman, C.J., Vega-Redondo, F.: Can social contagion spread without key players? In: BESC (2016)
16. Leskovec, J., Krevl, A.: SNAP Datasets: stanford large network dataset collection, June 2014. <http://snap.stanford.edu/data>
17. Romero, D., Meeder, B., Kleinberg, J.: Differences in the mechanics of information diffusion. In: WWW (2011)
18. Sun, E., Rosenn, I., Marlow, C.A., Lento, T.M.: Gesundheit! modeling contagion through Facebook news feed. In: ICWSM (2009)
19. Tong, H., Prakash, B., Eliassi-Rad, T., Faloutsos, M., Faloutsos, C.: Gelling, and melting, large graphs by edge manipulation. In: CIKM, pp. 245–254 (2012)
20. Viswanath, B., Mislove, A., Cha, M., Gummadi, K.P.: On the evolution of user interaction in Facebook. In: WOSN, August 2009
21. Watts, D.: A simple model of global cascades on random networks. *Proc. Nat. Acad. Sci. (PNAS)* **99**(9), 5766–5771 (2002)



Using Link Clustering to Detect Influential Spreaders

Simon Krukowski¹(✉) and Tobias Hecking²

¹ University of Duisburg-Essen, Forsthausweg 2, 47057 Duisburg, Germany
simon.krukowski@stud.uni-due.de

² German Aerospace Center, Linder Höhe, 51147 Cologne, Germany

Abstract. Spreading processes are increasingly analysed in the context of complex networks, for example in epidemics research, information dissemination or rumors. In these contexts, the effect of structural properties that facilitate or decelerate spreading processes are of high interest, since this allows insights into the extent to which those processes are controllable and predictable. In social networks, actors usually participate in different densely connected social groups that emerge from various social contexts, such as workplace, interests, etc. In this paper, it is examined if the number of groups an actor connects to can be used as a predictor for its capability to spread information effectively. The social contexts (i.e. groups) a node participates in are determined by the Link Communities approach by Ahn et al. (2010). The results are contrasted to previous findings of structural node properties based on the k -shell index of nodes (Kitsak et al. 2010) by applying both methods on artificially generated and real-world networks. They show that the approach is comparable to existing ones using structural node properties as a predictor, yet no clear evidence is found indicating that one or the other approach leads to better predictions in all investigated networks.

Keywords: Link clustering · Spreading processes · Information diffusion

1 Introduction

Spreading processes, originally examined in areas such as disease modelling [1, 2] and epidemiological mathematics [3], are increasingly examined to study social phenomena of information diffusion within complex networks, such as the spreading of rumours [4] or the communication during crisis events, for example [5]. They also gain special relevance considering the global COVID-19 pandemic, possibly yielding results to better understand and mitigate its spread. As a result of the way users connect and interact with each other, the social networks used for these analyses often exhibit properties of small-world and scale-free networks [6], making topological characteristics of the network an important aspect when analysing spreading processes. A common goal of the mentioned studies is to predict the efficiency of the spreading process, with the broader intention to acquire knowledge on how to control it. In this context, both local and global network properties related to spreading processes have to be explored. This paper focuses on local

properties (topological properties of the network attributed to nodes) but goes beyond their immediate neighbourhood. Using local properties to predict spreading efficiency, Kitsak et al. [7] already showed that the most efficient spreaders within a network are not necessarily the most connected nodes (i.e. nodes with the highest degree), but the ones that are located in densely connected cores of the network indicated by a high k -shell index. However, apart from being located within the core of a network or having a certain degree, structural properties of the network such as its tendency to form clusters might also yield an informative measure to predict the spreading efficiency. The general idea is, that someone who is member of many different overlapping social groups (workplace, sports club, friendship circles) is better capable of injecting information into various densely connected regions of the network where it further circulates. This notion of overlapping social groups is operationalised by applying the link-clustering approach by Ahn, Bagrow & Lehmann [8], where resulting clusters are highly interleaved and sometimes even nested. The approach clusters the links instead of the nodes, resulting in nodes possibly belonging to multiple clusters. It is reasonable to assume then, that the number of groups a node belongs to predicts its spreading capability as good as or even better than its k -shell index. Thus, we derive the following research questions:

- RQ1:** Is the community membership of nodes as determined by the Link Communities approach a good predictor for efficient spreaders within complex networks?
- RQ2:** Are the two approaches (Link Communities and k -Core) comparable for determining influential spreaders?

2 Background

Spreading processes and the analysis of potential spreaders have a long history in science and generally describe a flow of *information* between actors or members of a network [9]. For complex networks such as computer networks or networks of real individuals, *information* can refer to diseases and computer viruses [1, 10], whereas for other networks (i.e. created from Social Media data), it can refer to opinions, news articles or influence [11–13]. The spreading, i.e. the flow of *information* within these networks can result in diverse operationalisations, and obviously in contrary motifs regarding its analysis. For disease spreading, potential strategies to mitigate are sought-after, whereas for influence maximisation or opinion spreading, strategies to accelerate the flow of information are desired. For that reason, influencing factors are of great interest. Within complex networks where spreading can only happen between adjacent nodes, there is one aspect that affects the spreading, and likewise the efficiency of a single spreader – regardless of the motifs of analysis – to an equal degree: the topology of the network (see [9]). With regard to this topology, the origin of the diffusion process (i.e. the spreader) is of interest, as these so called “seeds” [14] and their properties yield important information from which inferences regarding the efficiency of the spreading process can be drawn.

2.1 Properties of the Network

As described above, the topology of a network results in certain characteristics of nodes, from which inferences regarding their spreading capability can be drawn. On an individual level, the degree centrality of a node is one such characteristic, as nodes with high degree centralities naturally have more possibilities to potentially spread information to other nodes [15]. Thus, so called “hubs” mark efficient spreaders (see [7]), which is also reflected by the fact that an uneven degree distribution (many hubs) results in more efficient spreading [7]. Apart from this, the community structure of a network can also influence spreading processes, as it is conceivable that information can spread more easily within highly interconnected sub-communities [16].

One notion to describe the community structure and the cohesiveness of subgraphs is the k -core of a network and the respective k -shell index of a node, which is the highest k such that the node is still part of the respective k -core. The index results in nodes which have at least k connections to other nodes within their core, resulting in highly interconnected nodes, whose spreading capability is high and where spreading is likely to occur. Furthermore, community detection techniques such as the Louvain method [17] can be used to examine the community structure of a network. However, as it assigns sub-communities to nodes based on high connections within a community and little connections between different sub-communities, each node is assigned a unique sub-community. To infer the spreading capability of a node however, its membership of a subcommunity yields little information: Information that originates from a spreader located within a highly interrelated sub-community might propagate quickly within the respective sub-community but is less likely to propagate to other sub-communities in the case of highly separated communities. In contrast, nodes with many different community memberships, indicating activities in various social contexts, are hypothesised to be capable spreaders. Such multiple community memberships can be found by clustering methods such as Clique Percolation [18] or Link Communities [8] that allow for overlapping sub-communities.

3 Approach

In this paper, it is hypothesised that actors who connect groups in different social contexts and thus are part of different overlapping and nested link communities are capable spreaders. The underlying assumption is that information items, diseases, etc. mainly circulate within densely connected groups. Actors in the overlap of such groups can be infected within one group and inject the spreading process into several other groups. To this end, we investigate whether the membership in multiple link communities (see Sect. 3.1) is another factor that determines spreading capability in addition to the node’s k -shell index [7]. It is further argued, that the number of link communities a node belongs to, also constituting a topological feature of the network organisation, suggests that the spreader has close connections to many other actors from different sub-communities within the network, and is thus able to spread between different highly interconnected communities more easily. Kitsak et al. [7] argue, that especially during the early stages of spreading processes, through the many pathways that exist for nodes located within the core of the network, the k -Shell index of a node predicts its spreading capability. However, nodes

within these cores also tend to exhibit multiple community memberships, which yields an additional inferential value in comparison to solely taking the node's k -Shell index into account (see Fig. 1). Additionally, when multiple outbreaks happen, information can spread more easily between different sub-communities, whereas for different cores, the distance between them needs to be taken into account [7]. The Link Communities approach by Ahn, Bagrow, & Lehmann [8] offers a method to determine overlapping communities, as it assigns multiple community-memberships to a single node. From this, inferences regarding the spreading capability of single nodes can be drawn.

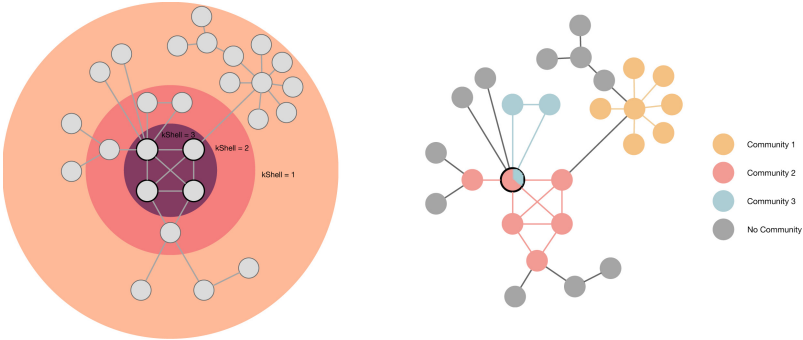


Fig. 1. On the left, an example from Kitsak et al. (2010) can be seen. Nodes within the core of the network, i.e. with a high k -Shell index, were found to be good spreaders. On the right, the same network is shown, but clustered with the Link Communities approach by Ahn et al. (2010). Nodes with black borders are nodes which are hypothesized to exhibit a high spreading capability. In this example, nodes with multiple community memberships also exhibit a high community centrality.

3.1 Link Communities

To determine the community-memberships of nodes, the Link Communities approach by Ahn et al. (2010) is used. In their approach, a sub-community is characterised as a set of closely interrelated links instead of closely interconnected nodes. As a result, sub-communities can overlap, and single nodes can be members of multiple sub-communities. The procedure to cluster the links by Ahn et al. (2010) is described as follows: Edges (e_{ik} and e_{jk}) with a common neighbour k are compared pairwise. Node k is called keystone node, while the other two nodes are called impost nodes. It should be noted, that

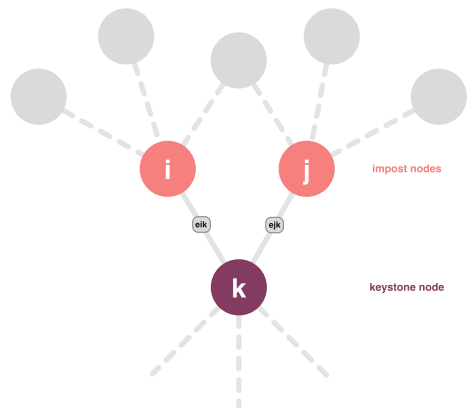


Fig. 2. Illustration from Ahn et al. (2010). As can be seen, only the neighbourhood of the impost nodes is taken into account.

only the neighbours of the impost nodes are taken into account for the calculation, as the neighbours of k (except the impost nodes) are of no interest. To calculate the similarity of the nodes, the similarity criterion S (Jaccard-index, [19]) is applied (see Eq. 1). The set of the node i and its neighbours is denoted as $n + i$.

$$S(e_{ik}, e_{jk}) = \frac{|n + (i) \cap n + (j)|}{|n + (i) \cup n + (j)|} \quad (1)$$

For the above example (Fig. 2), this would result in $S = \frac{1}{4}$. A dendrogram is then built through single-linkage hierarchical clustering and cut at a certain threshold according to the partition density, which then results in the link communities [8]. From these link communities, the community memberships of the nodes can be derived, and thus each node is assigned a vector of community memberships, from which the actual number of communities it belongs to can be calculated. The possibility to detect multiple community memberships differentiates our approach from other approaches analysing properties of influential spreaders [20], where nodes can only belong to one community each due to the community detection algorithm being used.

Community Centrality. Although the number of communities a node belongs to might be an important predictor for its spreading capability, there are possible limitations. Generally, due to the nested nature of link communities, a node can be a member of many communities. One can assume, that being a member of many communities goes along with a high spreading capability. However, because of this pervasive overlap, the set of directly reachable nodes can be limited, even for nodes belonging to multiple groups and the inferential value of the nodes' number of community-memberships might be limited. This is reflected by the community centrality by Newman [21], which assigns higher centrality values to nodes if they belong to many communities with little overlap. In this study, this concept is extended, additionally taking the size of the communities into account. For simplicity, it will also be denoted as community centrality. Formally, it is defined as the cardinality of the union of nodes in all communities a node belongs to. Consequently, it is high if a node belongs to many large communities with little overlap. Community centrality will be denoted as CC.

4 Evaluation

To evaluate the capability of nodes to spread information through the network, spreading processes are simulated according to well-known SIR models [3]. The process starts with one initially infected node. This node infects its neighbours at a given infection rate (denoted as β) and recovers. The resulting infected nodes then try to infect their neighbours themselves. The process terminates when no new infections occur. In this study, the spreading capability of a node x corresponds to the average fraction of infected population in 100 SIR runs starting at node x . For the community detection, it is decided to cut the resulting dendrogram at a smaller threshold to also detect smaller sub-communities.

Analysed Datasets. To evaluate our measure, we chose to use both real-world networks to increase the external validity, as well as artificially generated networks to maintain a high internal validity. The generated networks were created according to the forest-fire algorithm as shown by Leskovec, Kleinberg & Faloutsos [22], as it creates networks with properties typical of real-world graphs such as heavy-tailed degree distributions and communities. To this end, 8 undirected networks were created, each with 1000 nodes, with forward burning probabilities ($fwprob$) ranging from .05 to .40 (Table 1) and a fixed backward burning probability of 1 (see [22]). The $fwprob$ controls for the tendency to form densely connected and potentially nested clusters. Additionally, we analysed one ego-networks from the SNAP (Stanford Network Analysis Project) at Stanford University [23] to evaluate our metric on real-world data. The network represents the connections between all friends of the individual of which the ego network is derived from (thus *ego*), with all connections between the ego and friends removed. To increase informative value and evaluate our measure on a bigger network than our created ones, we chose the *107* network because of its size (at 1912 nodes and 53498 edges), a local average clustering coefficient of .534 and a mean spreading capability of .683.

Table 1. Analysed datasets. We used the average local clustering coefficient. The spreading capability is the mean spreading capability of all nodes.

Network	Edges	avg. Degree	Clust. Coeff.	Diameter	Spreading Cap.	FwProb
1	1,076	2.152	0.160	23	0.066	0.050
2	1,193	2.386	0.306	24	0.136	0.100
3	1,275	2.550	0.358	21	0.236	0.150
4	1,416	2.832	0.385	21	0.339	0.200
5	1,829	3.658	0.490	19	0.552	0.250
6	2,271	4.542	0.530	16	0.638	0.300
7	5,455	10.910	0.540	11	0.792	0.350
8	48,355	96.710	0.837	8	0.952	0.400

4.1 Metrics for Evaluation

In addition to descriptively comparing the proposed measure with established measures, certain metrics are applied to objectively evaluate it.

The Imprecision Function. To quantify the importance of nodes with a high community centrality during the spreading, an objective measure has to be calculated. The imprecision function serves just that purpose. Similar to Kitsak et al. (2010), these functions are calculated for each of the three relevant measures, and they are denoted as $\varepsilon_{k_S}(p)$, $\varepsilon_{CC}(p)$ and $\varepsilon_d(p)$, respectively. For each subset p of nodes (here, p refers to a specific percentage of the dataset) with the highest spreading capability (denoted as ϕ_{eff}) and the highest value according to one of the three measures (denoted as φ_{k_S} , φ_{CC} and φ_d), the average spreading is calculated. Then, the difference in spreading between the

p nodes with highest values in any of the measures and the most efficient spreaders is calculated. Formally, for ϵ_{CC} , the function is defined as follows:

$$\epsilon_{CC}(p) = 1 - \frac{\varphi_{CC}(p)}{\varphi_{eff}(p)} \quad (2)$$

Note that through subtracting the fraction from 1, higher values correspond to more imprecision, while smaller values for ϵ show less imprecision and thus a better measure.

4.2 Results

In the following, the results of the evaluation will be presented. All of the evaluations are calculated at a beta value (infection rate) of 7%.

Descriptive Results

General Observations. To assess how well the community memberships of a node allow inferences about its spreading capability, they are compared to other measures, more specifically to the k -Shell index of a node [7] and to its degree. As can be seen in Fig. 3, the measures generally correspond to each other, and higher $fwprob$ values result in higher values for CC, degree and k -Shell index. Figure 4 shows the results of a bivariate comparison for our generated networks with 1000 nodes, beta = 7 and $fwprob$ values of .05, .15 and .30. It can be seen that higher $fwprob$ values correspond to higher spreading capabilities, likely because of more community structure within the graphs. While the predictive value of all shown measures seems to be smaller for lower $fwprob$ values with regard to the spreading capability, it increases for higher $fwprob$ values. The figure shows, that generally, the CC corresponds with the degree and the k -Shell index of a node. Especially in comparison with the degree of a node, it can be seen that low degree values do not consistently correlate with low spreading capabilities, whereas for the CC, they do. For $fwprob = .30$, higher CC values also consistently reflect high spreading capabilities, whereas for degree, there are effective spreaders with small degrees. However, this effect shows less so for the k -Shell index of a node, where effective spreaders can be found for a greater variety of CC values, especially at high k -Shell values. Regarding **RQ1**, it can be said that the CC can also be used as a predictor for the spreading capability of the nodes: For high CC values, there are high spreading capabilities whereas for small CC values, spreading capabilities generally remain low. This discriminatory value increases with higher $fwprob$ values and thus higher community structures within the

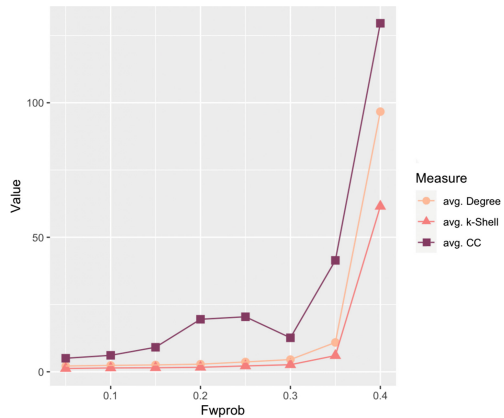


Fig. 3. Average values of the compared measures for the chosen $fwprob$ values.

graphs. To additionally evaluate this on a real network, we created the same plot for the *107* network (Fig. 5) at $\beta = 1$ (high β values would result in little variance of the spreading capability), where a similar pattern as above and in Kitsak et al. [7] emerges. Even more clearly than for the generated networks, it can be seen that high values in either community centrality, degree or k -Shell index, go along with a high spreading capability. Thus, and also considering what can be seen in Fig. 3, both the k -Shell index and the Link Communities approach are comparable in predicting the spreading capability (RQ2). However, this effect shows less so for our generated networks at high $fwprob$ values than for our analysed *107* network.

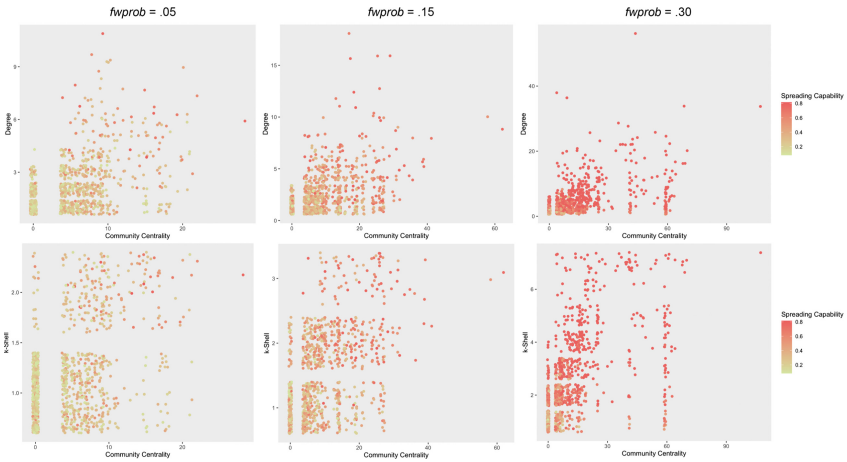


Fig. 4. Bivariate distribution for the examined measures on our generated networks. The colours indicate the spreading capability.

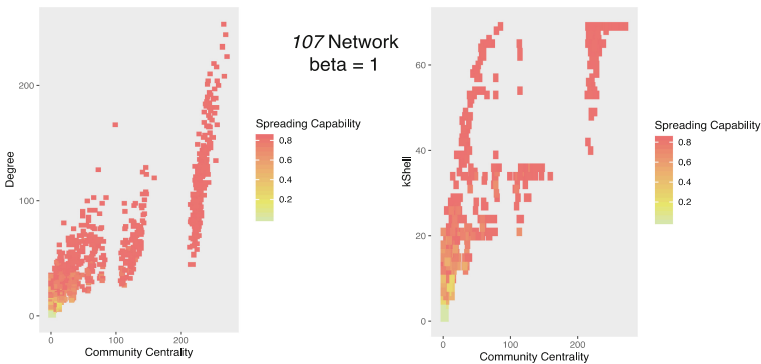


Fig. 5. Bivariate distribution for the 107 Facebook ego network

Correlational Measures. Correlations between the CC and the established measures are calculated. The mean correlation across all of our generated networks between the CC

and the k -Shell index of a node is $r = .56$ and $r = .54$ between CC and the degree. All measures correlate with the spreading capability of a node ($r = .40$ for CC, $r = .40$ for degree and $r = .47$ for k -Shell). In the *107* network, the CC also correlates with the other measure ($r = .90$ with k -Shell and $r = .88$ for degree) and with the spreading capability of a node (r values at $.50$ for CC, $.55$ for degree and $.63$ for k -Shell). Regarding **RQ2**, this makes the CC comparable to the k -Shell index of a node.

Imprecision Function. Apart from using correlational measures and looking at the distribution of data points, focusing on the top n nodes, the imprecision function is applied to all of the studied networks, in order to objectively evaluate the proposed measure in comparison to the other measures. It was calculated for all of our generated networks at $\beta = 7$. As can be seen in Fig. 6 very clearly, the errors decrease with higher $fwprob$ values, reflecting the observation described above and indicating that the predictive value of all measures (the CC included) seems to increase when there is a higher community structure. However, with $M = 0.18$ ($SD = 0.20$) for the k -Shell index, $M = 0.10$ ($SD = 0.12$) for degree and $M = 0.15$ ($SD = 0.20$) for community centrality, they are generally quite low, with the k -Shell index showing the highest average error. For the *107* network, the error values are especially low at $.01$ for degree, $.02$ for k -Shell and $.01$ for the CC. As there is no significant difference between the imprecision of our measure and the imprecision of other measures for both real and generated networks, the measures are comparable (**RQ2**), while the low error values additionally indicate the CC to be a good predictor for the spreading capability of a node (**RQ1**).

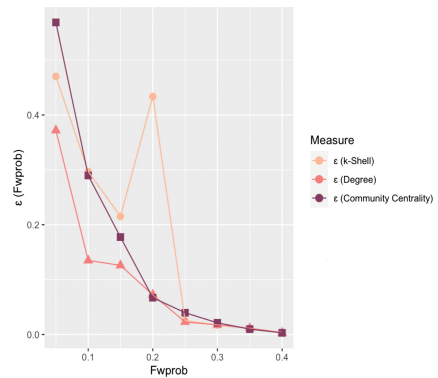


Fig. 6. Results of the imprecision function for our generated networks.

5 Discussion

The aim of this paper is to contribute to research on spreading processes within complex networks and identify properties which increase the efficiency of spreading. To this end, a novel measure is introduced, from which inferences regarding the spreading capability of single actors can potentially be drawn. Using link clustering [8] to infer multiple community memberships of nodes, we evaluate and compare the measure to other approaches. Generally, our results extend the understanding of the effect of structural properties of networks on information diffusion beyond centralities and k -shell and show that the community centrality of a node is comparable in predicting its spreading capability.

Descriptively, the average values of the measures indicate a comparability (**RQ2**): Higher $fwprob$ values and thus a higher community structure go along with higher k -Shell indices, degrees and community centrality. The bivariate plots (Fig. 4) extend this

with regard to predicting effective spreaders, additionally hinting at a broad range of degrees for a small range of community centralities. While the correlation shows less so in comparison with the k -Shell indices for our generated networks, it does show for the observed *107* network. There, high community centralities reflect less variance in spreading capability, implying a more robust prediction. This is especially true for the comparison between the CC and the k -Shell index, making it a good predictor in our analysed real network (**RQ1**). The calculated correlations imply the comparability of our measure, as it correlates with the other measures and also shows a medium-high correlation with the spreading capability of a node - for both generated and real networks. Judging from these correlations alone, regarding **RQ1**, this does also make the CC a good predictor for effective spreaders, yet slightly less than the k -Shell index. However, it should be noted that the expressiveness of correlations is limited in this case because of long-tailed and different distributions of the variables. The high correlations between the variables underline this further, as for example, our generated networks are scale free with many nodes showing a low degree, and consequently, also a low community centrality. Thus, in addition, the imprecision function focusing on the top n nodes is evaluated. The results clearly show that the errors in predicting the most efficient spreaders decrease when there are higher *fwprob* values and thus more community structure – for the CC along with the other analysed measures. This is especially relevant, as it shows that with regard to the results of the imprecision function, our measure is not only a good predictor for efficient spreaders (**RQ1**), yet this predictive value increases when the networks show more properties of real world networks with communities and heavy-tailed degree distributions. This is also reflected by the results for the *107* network and additionally undermined by the generally small error values. In our analysed networks, error values were even slightly higher for the k -Shell index than for the CC, exceeding our **RQ2** of comparability. In conclusion, the evaluations of the proposed measure show its comparability to other measures, specifically the degree and the k -Shell index of a node.

Certain things should be taken into account. For our generated networks with high *fwprob* values and the *107* network, the mean spreading capability of the sample is very high. This extreme right-skewness means, that many nodes show a high spreading capability, and takes away possibilities to examine properties that lead to such high spreading capability. Apart from that, due to computational constraints, the possibilities of simulating the spreading processed in the networks were limited, resulting in capping the size at 1000 nodes. Additionally, *fwprob* values greater than .40 resulted in networks with more than 100,000 edges, also making the simulations computationally intensive. It might therefore be possible that bigger networks or higher *fwprob* values show different results – calling for future analyses with bigger networks. For our chosen real network, its characteristics might have also influenced the metrics used for the conducted analyses. Thus, other networks (real-world, computer-networks or networks with ground-truth communities) should also be used for future analyses.

While we systematise the degree of real-world properties of the generated networks by varying the *fwprob* values with which they are created, we do not systematise all aspects of our analyses: There are aspects of our Julia implementation which we use to simulate the spreading that are fixed, specifically the steps (fixed to 30) and the iterations

(fixed to 10), and likewise the beta value at .7 (for the generated networks). Future studies should also vary these fixed parameters and try to systematise them like the systematised *fwprob* values in this paper. Additionally, the spreading capabilities obtained are also not deterministic, that means they rely heavily on chance. It is therefore possible that another run yields slightly different results.

Conclusion and Outlook. Our analyses and evaluations showed, that apart from the *k*-Shell index and centralities, structural properties of the network can affect spreading processes. To this end, community centrality, the examined measure, proved to be comparable in doing so. Along with the other analysed measures, the CC's inferential value increases, as the *fwprob* used to create the network with the forest fire algorithm [21] increases – meaning that for increasing real-world and scale free properties of a graph, the CC becomes better in predicting efficient spreaders. While the *k*-Shell index of a node seems to be a better predictor for the spreading capability under certain conditions, there might be applications in which the *k*-Shell index yields little inferential value or where there are multiple outbreaks simultaneously. In this case, the community centrality might be used instead of the *k*-Shell index coupled with the distance between cores [7]. This paper contributes to our understanding of the underlying processes through offering another measure that can be used to infer the spreading capability of nodes, and thus the efficiency of information diffusion due to structural properties of complex networks. Due to factors that could have influenced the evaluations in the present paper, future studies should further evaluate the measure, apply it to different contexts and networks and possibly apply new evaluation metrics.

References

1. Goltsev, A.V., Dorogovtsev, S.N., Oliveira, J.G., Mendes, J.F.F.: Localization and spreading of diseases in complex networks. *Phys. Rev. Lett.* **109**(12), 128702 (2012). <https://doi.org/10.1103/PhysRevLett.109.128702>
2. Kong, X., Qi, Y., Song, X., Shen, G.: Modeling disease spreading on complex networks. *Comput. Sci. Inf. Syst.* **8**(4), 1129–1141 (2011)
3. Keeling, M.J., Rohani, P.: *Modeling Infectious Diseases in Humans and Animals*. Princeton University Press (2011)
4. Qian, Z., Tang, S., Zhang, X., Zheng, Z.: The independent spreaders involved SIR rumor model in complex networks. *Phys. Stat. Mech. Its Appl.* **429**, 95–102 (2015). <https://doi.org/10.1016/j.physa.2015.02.022>
5. Stieglitz, S., Bunker, D., Mirbabaie, M., Ehnis, C.: Sense-making in social media during extreme events. *J. Contingencies Crisis Manag.* **26**(1), 4–15 (2018). <https://doi.org/10.1111/1468-5973.12193>
6. Wu, F., Huberman, B.A., Adamic, L.A., Tyler, J.R.: Information flow in social groups. *Phys. Stat. Mech. Appl.* **337**(1), 327–335 (2004). <https://doi.org/10.1016/j.physa.2004.01.030>
7. Kitsak, M., et al.: Identification of influential spreaders in complex networks. *Nat. Phys.* **6**(11), 888–893 (2010). <https://doi.org/10.1038/nphys1746>
8. Ahn, Y.-Y., Bagrow, J.P., Lehmann, S.: Link communities reveal multiscale complexity in networks. *Nature* **466**, 761 (2010)
9. Karunakaran, R.K., Manuel, S., Narayanan Satheesh, E.: Spreading information in complex networks: an overview and some modified methods. In: *Graph Theory - Advanced Algorithms Applications*, December 2017. <https://doi.org/10.5772/intechopen.69204>

10. Liu, Q., Han, L., Zou, M., Wei, O.: Spread dynamics with resistances of computer virus on complex networks. In: 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), vol. 2, pp. V2-27–V2-31, August 2010. <https://doi.org/10.1109/icacte.2010.5579087>
11. Törnberg, P.: Echo chambers and viral misinformation: modeling fake news as complex contagion. *PLoS ONE* **13**(9), e0203958 (2018). <https://doi.org/10.1371/journal.pone.0203958>
12. Vosoughi, S., Roy, D., Aral, S.: The spread of true and false news online. *Science* **359**(6380), 1146–1151 (2018). <https://doi.org/10.1126/science.aap9559>
13. Alvarez-Galvez, J.: Network models of minority opinion spreading: using agent-based modeling to study possible scenarios of social contagion. *Soc. Sci. Comput. Rev.* **34**(5), 567–581 (2016). <https://doi.org/10.1177/0894439315605607>
14. Comin, C.H., da Fontoura Costa, L.: Identifying the starting point of a spreading process in complex networks. *Phys. Rev. E* **84**(5), 056105 (2011). <https://doi.org/10.1103/physreve.84.056105>
15. Albert, R., Jeong, H., Barabási, A.-L.: Error and attack tolerance of complex networks. *Nature* **406**(6794), 378–382 (2000). <https://doi.org/10.1038/35019019>
16. Stegehuis, C., van der Hofstad, R., van Leeuwen, J.S.H.: Epidemic spreading on complex networks with community structures. *Sci. Rep.* **6**, 29748 (2016). <https://doi.org/10.1038/srep29748>
17. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech: Theory Exp.* **2008**(10), P10008 (2008). <https://doi.org/10.1088/1742-5468/2008/10/P10008>
18. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**(7043), 814–818 (2005). <https://doi.org/10.1038/nature03607>
19. Jaccard, P.: The distribution of the flora in the alpine zone. *New Phytol.* **11**(2), 37–50 (1912). <https://doi.org/10.1111/j.1469-8137.1912.tb05611.x>
20. Hu, Q., Gao, Y., Ma, P., Yin, Y., Zhang, Y., Xing, C.: A new approach to identify influential spreaders in complex networks. In: *Web-Age Information Management*, pp. 99–104, Berlin, Heidelberg, (2013). http://doi.org/10.1007/978-3-642-38562-9_10
21. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* **74**(3), 036104 (2006). <https://doi.org/10.1103/PhysRevE.74.036104>
22. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, **1**(1), 2–es (2007). <https://doi.org/10.1145/1217299.1217301>
23. Leskovec, J., McAuley, J.J.: Learning to discover social circles in ego networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems*, vol. 25, Curran Associates, Inc. pp. 539–547 (2012)



Prediction of the Effects of Epidemic Spreading with Graph Neural Networks

Sebastian Mežnar¹, Nada Lavrač^{1,2}, and Blaž Škrlj^{1,2}(✉)

¹ Jožef Stefan Institute, Jamova 39, Ljubljana, Slovenia
blaz.skrlj@ijs.si

² Jožef Stefan International Postgraduate School, Jamova 39, Ljubljana, Slovenia

Abstract. Understanding how information propagates in real-life complex networks yields a better understanding of dynamical processes such as misinformation or epidemic spreading. With the recent resurgence of graph neural networks as a powerful predictive methodology, many network properties can be studied in terms of their predictability and as such offer a novel view on the studied process, with the direct application of fast predictions that are complementary to resource-intensive simulations. We investigated whether graph neural networks can be used to predict the effect of an epidemic, should it start from a given individual (patient zero). We reformulate this problem as node regression and demonstrate the high utility of network-based machine learning for a better understanding of the spreading effects. By being able to predict the effect of a given individual being the patient zero, the proposed approach offers potentially orders of magnitude faster risk assessment and potentially aids the adopted epidemic spreading analysis techniques.

Keywords: Epidemics · Neural networks · Machine learning · Spreading

1 Introduction

The spread of information and disease is a common phenomenon that has a lot of practical and sometimes life-saving applications. One of these applications is the creation of better strategies for stopping the spreading of misinformation on social media or an epidemic. Further, companies can analyze spreading to create better strategies for marketing their product [6, 19]. Spreading analysis can also be suitable for analysis of e.g., fire spreading, implying large practical value in terms of insurance cost analysis [8]. Analysis of spreading is commonly studied via extensive simulations [11]. Here, the ideas from statistical mechanics are exploited to better understand both the extent of spreading, as well as its speed [2].

Albeit offering high utility, reliable simulations of spreading processes can be *expensive* on larger networks, which we believe can be addressed by the employment of *machine learning*-aided modelling [29]. The contributions of this work are multifold and can be summarized as follows.

1. We re-formulate the task of identification of the spreading effect from a given node into a node regression problem.
2. The prediction problem is addressed with state-of-the-art graph neural network-based approaches, as well as a simpler, centrality-based approach proposed as a part of this work.
3. Consistent predictive capability is demonstrated across multiple real-life networks, demonstrating that graph neural network-based approaches can serve as a complementary, highly efficient analysis tool when studying information spreading.
4. A methodology is proposed that performs notably better than the random baseline on the datasets we tested.
5. We demonstrate how individual predictions of the obtained models can be *explained* via the game-theoretic explanation tool SHAP [17].

The remainder of this work is structured as follows. In Sect. 2, we discuss the related work which led to the proposed approach. Next, we re-formulate the task and show its importance in Sect. 3. We propose a new approach based on node centralities to solve the re-formulated task in Sect. 4. In Sect. 5 we present the datasets, experimental setting and results of our empirical evaluation. We conclude the paper in Sect. 6.

2 Related Work

In the following section, we discuss the relevant related work. We begin by discussing the notion of contagion processes, followed by an overview of graph neural networks.

2.1 Analysis of Spreading Processes

The study of spreading processes on networks is a lively research endeavour [19]. Broadly, spreading processes can be split into two main branches, namely, the simulation of *epidemics* and *opinion dynamics*. The *epidemic spreading* models can be classical or network-based. Here, the classical models are for example systems of differential equations that do not account for a given network's topology. Throughout this work, we are interested in extensions of such models to real-life network settings. One of the most popular spreading models on networks is the Susceptible-Infected-Recovered (SIR) [10] model. The spread of the pandemic in the SIR model is dictated by parameters β known as the infection rate and γ known as the recovery rate. Nodes in this model can have one of three states (Susceptible, Infected, Recovered).

SIR assumes that if a susceptible node comes into contact with an infected one during a generic iteration, it becomes infected with probability β . In each iteration after getting infected, a node can recover with probability γ (the only transition allowed are S to I to R).

Other related models include, for example, SEIR, SEIS, SWIR¹. Further, one can also study the role of cascades [9] or the Threshold model [4].

¹ Where S-Susceptible, I-Infected, R-Recovered, E-Exposed and W-Weakened.

2.2 Machine Learning on Networks

Learning by propagating information throughout a given network has already been considered by the approaches such as label propagation [30]. However, in recent years, the approaches that jointly exploit both the adjacency structure of a given network alongside features assigned to its nodes are becoming prevalent in the network learning community. The so-called graph neural networks have re-surfaced with the introduction of the Graph Convolutional Networks (GCN) [13]; an idea where the normalized adjacency matrix is directly multiplied with the feature space and effectively represents a neural network layer. Multiple such layers could be stacked to obtain better approximation power/performance. One of the most recent methods from this branch are the Graph Attention Networks [28], an extension of GCNs with the notion of neural *attention*. Here, part of the neural network focuses on particular parts of the adjacency space, offering more robust and potentially better performance.

Albeit being in widespread use, graph neural networks are not necessarily the most suitable choice when learning solely from the network adjacency structure. For such tasks, methods such as node2vec [5], SGE [26] and DeepWalk [21] were developed. This branch of methods corresponds to what we refer to as *structural* representation learning. In our work, we focus mostly on learning this type of representations using network centrality information.

Note that although graph neural networks are becoming the prevailing methodology for learning from *feature-rich* complex networks, it is not clear whether they perform competitively to the more established structural methods if the feature space is derived solely from a given network's structure.

3 Task Formulation

When analysing an epidemic there are three main pieces of information that give us most insight about how severe an epidemic was. The first crucial information is when an epidemic reaches the peak (most nodes infected) since we are less likely to be able to stop an epidemic when the peak is reached too quickly. This information is especially important when trying to find a cure for a disease or trying to stop misinformation on social media. Related to this, we usually want to know, how many nodes will be infected when the epidemic reaches its peak. When the maximum number of people infected by some disease is high, there might not be enough beds or medicine for everyone. In contrast, companies might want to create marketing campaigns on platforms such as Twitter and target specific users to reach a certain number of retweets that are needed to become trending. Another important insight into an epidemic is how many people get infected. If a scam on the internet reaches a lot of people there is a greater chance that more people will fall for it.

In our work, we focus on predicting the maximum number of infected nodes and the time this number is reached. We create target data by simulating epidemics from each node with SIR diffusion model and identifying the number, as well as the time when the maximum number of nodes are infected. We aggregate

the generated target data by taking the mean values for each node. In the end, we preprocess this data by normalizing it.

4 Proposed Methodology

In this section, we present the creation of target data and summarize centralities and learners used for the regression task. An overview of the proposed methodology can be seen in Fig. 1. The figure shows two branches. On the upper branch, simulations are created and transformed into target data, while the node representation is learned on the lower branch. After this, a regression model is trained with data from both branches and used to generate predictions for the remaining (unknown) nodes.

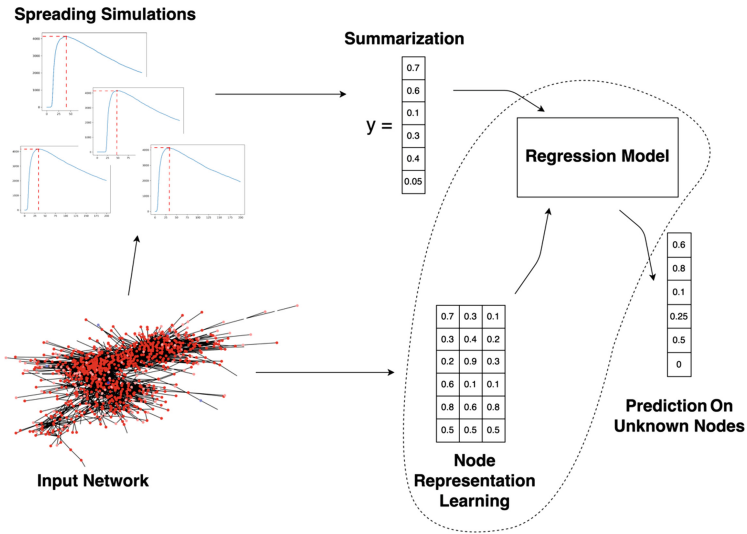


Fig. 1. Overview of the proposed methodology.

The initial part of the methodology addresses the issue of input data generation. Intuitively, the first step simulates epidemic spreading from individual nodes of a given network to assess both the time required to reach the maximum number of infected, as well as the number itself. In this work, we leverage the widely used SIR model [10] to simulate epidemics, formulated as follows.

$$\begin{aligned} \frac{dS}{dt} &= -\frac{\beta \cdot S \cdot I}{N} \\ \frac{dI}{dt} &= \frac{\beta \cdot S \cdot I}{N} - \gamma \cdot I \\ \frac{dR}{dt} &= \gamma \cdot I, \end{aligned}$$

where S represents the number of susceptible, R the number of recovered and I the number of infected individuals. Spreading is governed by input parameters γ and β . The SIR model is selected due to many existing and optimized implementations that are adapted from systems of differential equations to networks [6]. We use NDlib [23] to simulate epidemics based on the *SIR diffusion model*.

Target data creation results in two real values for each node. We attempt to *predict* these two values. The rationale for the construction of such predictive models is, they are potentially much faster than simulating multiple processes for each node (prediction time is the bottleneck) and can give more insight into *why* some nodes are more “dangerous”. The predictive task can be formulated as follows. Let $G = (V, E)$ represent the considered network. We are interested in finding the mapping $f : V \rightarrow \mathbb{R}^+$, such that this mapping maps from the set of nodes V to the set of real values that represent e.g., the maximum number of infected individuals if the spreading process is started from a given node $u \in V$. Thus, f corresponds to **node regression**.

The models we considered can broadly be split into two main categories; graph neural networks and propositional learners. The main difference between the two is that the graph neural network learners, such as GAT [28] and GIN [31] simultaneously exploit both structure of a network, as well as *node features*, whilst the propositional learners take as input only the constructed feature space (and not the adjacency matrix). As an example, the feature space is passed throughout the GIN’s structure via the update rule that can be stated as:

$$\mathbf{h}_v^{(k)} = \text{MLP}^{(k)} \left((1 + \epsilon^{(k)}) \cdot \mathbf{h}_v^{(k-1)} + \sum_{u \in V(v)} \mathbf{h}_u^{(k-1)} \right),$$

where MLP corresponds to a multilayer perceptron, ϵ a hyperparameter, $\mathbf{h}_u^{(k)}$ the node u ’s representations at layer k and $V(v)$ the v -th node’s neighbors. We test both graph neural networks and propositional learners as it is to our knowledge not clear, whether direct incorporation of the adjacency matrix offers any superior performance, as the graph neural network models are computationally more expensive. The summary of considered learners is offered in Table 1.

As the considered complex networks do not possess *node attributes*, we next discuss which features, derived solely from network structure were used in the considered, state-of-the-art implementations of GAT [28] and GIN [31]. Further, we also test models where only the constructed structural features are considered, as well a standalone method capable of learning node representations, combined with the XGBoost [1] classifier. In this work, we explored whether *centrality-based* descriptions of nodes are suitable for the considered learning task. The rationale for selecting such features is, they are potentially fast to compute and entail global relation of a given node w.r.t. the remaining part of the networks. The centralities, computed for each node are summarized in Table 2. After calculating these centralities, we normalize and concatenate them to create the final features. These features together with XGBoost classifier are referred to as CABoost in Sect. 5.3, which is considered one of the contributions of this work.

Table 1. Summary of the considered learners with descriptions. Here, \mathbf{A} denotes the adjacency matrix and \mathbf{F} the feature matrix.

Input	Learner	Method description
\mathbf{A}, \mathbf{F}	GAT	Graph Attention Networks
\mathbf{A}, \mathbf{F}	GIN	Graph Isomorphism Networks
\mathbf{A}	node2vec + XGBoost	node2vec-based features + XGBoost
\mathbf{F}	CABoost (our)	XGBoost trained solely on centrality based features

Table 2. Summary of the centralities considered in our work.

Centrality	Time complexity	Description
Degree centrality [22]	$\mathcal{O}(E)$	The number of edges of a given node
Eigenvector centrality [22]	$\mathcal{O}(V ^3)$	Importance of the node, where nodes are more important if they are connected to other important nodes. This can be calculated using the eigenvectors of the adjacency matrix
PageRank [20]	$\mathcal{O}(E)$	Probability that a random walker is at a given node
Average Out-degree	$\mathcal{O}(V \cdot w \cdot \bar{s})$	The average out-degree of nodes encountered during w random walks of mean length \bar{s}
Hubs and Authorities (HITS) [14]	$\mathcal{O}(E)$	HITS is a link analysis algorithm that assigns two scores to each node. Authority score represents how important a node is and the hub score represents how well a node is connected to other important nodes

During model training we minimized the mean squared error between the prediction ($f(u)$) and the ground truth (y_u); stated for the u -th node as

$$MSE = \frac{1}{|N|} \sum_{u \in N} (f(u) - y_u)^2.$$

To summarize, we learn network features with fast algorithms and use them together with GIN, GAT and XGBoost learners to minimize the mean squared error between predictions and data we make using simulations on part of the network. We next present the evaluation process and results of this methodology.

5 Empirical Evaluation

In this section, we show the empirical results of our approach and compare it to other baselines. We also present how predictions from CABoost model can be explained using SHAP [16].

5.1 Baselines

We compared the results of proposed method to the following 4 baselines:

- *Random baseline* creates an embedding of size $|N| \times 64$ with random numbers drawn from $\text{Unif}(0, 1)$.
- *node2vec* [5] learns a low dimensional representation of nodes that maximizes the likelihood of neighborhood preservation using random walks. During testing, we use the default parameters.
- *GAT* [28] includes attention mechanism that helps learn the importance of neighboring nodes. In our tests, we use the implementation from PyTorch Geometric [3].
- *GIN* [31] learns a representation that can provably achieve the maximum discriminative power. In our tests, we use the implementation from PyTorch Geometric [3].

For comparison we also add the averaged simulation error. We calculate this error with the MSE formula, where we use the mean absolute difference between the value we get from simulations and the mean value for that node as $f(u)$ and 0 as y_u . This baseline corresponds to a situation, where only a single simulation would be used to approximate the expected value of multiple ones (the goal of this work).

5.2 Experimental Setting

For testing², we used datasets: Hamsterster [7], Advogato [18], Wikipedia Vote [15] and FB Public Figs. [25], taken from the Network Repository website [24]. Some basic statistics of the networks we used can be seen in Table 3. Two networks used during testing are visualized in Fig. 2. The network nodes in this figure are colored based on the values of the target variables.

Table 3. Basic statistics of the networks used for testing.

Name	Nodes	Edges	Components
Wikipedia vote [15]	889	2914	1
Hamsterster [7]	2426	16630	148
Advogato [18]	6551	43427	1441
FB public figures [25]	11565	67114	1

We used the following approach to test the proposed method as well as baselines mentioned in Sect. 5.1. We created the target data by simulating five epidemics starting from each node of every dataset. We created each simulation

² That can be found at <https://github.com/smezmar/Epidemic-spreading-CN2020>.

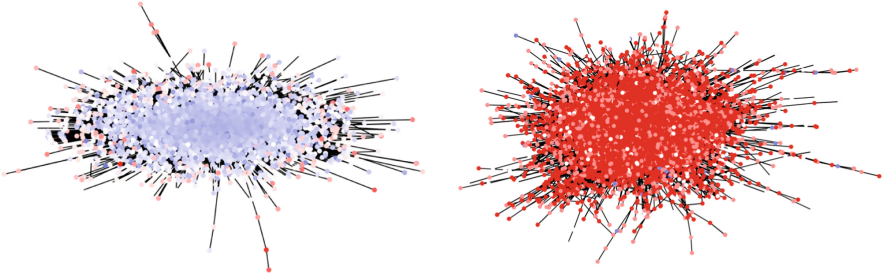


Fig. 2. Visualization of Advogato (left) and FB Public Figures (right) networks. The color represents the target value we get when starting the spreading from a given node. Color on Advogato dataset represents the time needed to reach the peak while on FB Public Figures dataset maximum number of infected nodes is shown. Blue colors represent low values while red ones represent high ones.

using the SIR diffusion model from the NDlib [23] Python library with parameters $\beta = 5\%$ and $\gamma = 0.5\%$. We then create the target variables by identifying and aggregating the maximum number of infected nodes and the time when this happens. We use these variables to test methods using five-fold cross-validation. We used XGBoost [1] with default parameters as the regression model with proposed features based on the mentions centralities, the random baseline and the node2vec [5] baseline. Baselines GIN and GAT were trained for 200 epochs using the Adam optimizer [12].

5.3 Results

The results of the evaluation described in Sect. 5.2 are presented in Tables 4 and 5. We can see that in most cases both time and the maximum number of infected nodes can be predicted significantly better by using information about the structure of the network.

Observing results in Table 4 we see that overall CABoost achieves best results and is beaten only on the Wiki vote dataset by GAT and the Hamsterster dataset by GIN. We can further see that although node2vec does not achieve the best score on any dataset, it consistently achieves results that are comparable to CABoost. The biggest improvement over the random baseline can be seen on datasets Hamsterster and Advogato that have more than one connected components. We can also see that only GAT achieves a result that is significantly better than the random baseline on the Wiki vote dataset.

The results in Table 5 are very similar to those of showcased in Table 4. Here CABoost performs even better and is outperformed only by GIN on the Wiki vote dataset. We can see that when predicting time, random baseline performs significantly worse than all other baselines on all datasets but that overall these predictions are better than when the maximum number of infected nodes is being predicted.

We can see on all datasets that prediction with such learners is more beneficial than creating only one simulation, further showing their usefulness.

Table 4. Cross-validation results for maximum number of infected node.

	Wiki vote	Hamsterster	Advogato	FB public figures
Random	0.0191 (± 0.0046)	0.1633 (± 0.0123)	0.2052 (± 0.0055)	0.0144 (± 0.0014)
node2vec+XGBoost	0.0200 (± 0.0034)	0.0060 (± 0.0019)	0.0073 (± 0.0010)	0.0127 (± 0.0009)
GAT	0.0149 (± 0.0039)	0.0460 (± 0.0015)	0.0653 (± 0.0079)	0.0129 (± 0.0009)
GIN	0.0234 (± 0.0078)	0.0042 (± 0.0006)	0.0253 (± 0.0195)	0.0116 (± 0.0007)
CABoost (our)	0.0187 (± 0.0040)	0.0045 (± 0.0012)	0.0067 (± 0.0012)	0.0114 (± 0.0011)
Simulation error (averaged)	0.0486 (± 0.0481)	0.0083 (± 0.0223)	0.0107 (± 0.0251)	0.0546 (± 0.0375)

Table 5. Cross-validation results for time when most nodes are infected.

	Wiki vote	Hamsterster	Advogato	FB public figures
Random	0.0168 (± 0.0015)	0.0168 (± 0.0013)	0.0390 (± 0.0014)	0.0094 (± 0.0004)
node2vec+XGBoost	0.0126 (± 0.0010)	0.0062 (± 0.0005)	0.0053 (± 0.0007)	0.0054 (± 0.0004)
GAT	0.0118 (± 0.0010)	0.0125 (± 0.0008)	0.0190 (± 0.0015)	0.0066 (± 0.0003)
GIN	0.0096 (± 0.0012)	0.0045 (± 0.0005)	0.0126 (± 0.0100)	0.0045 (± 0.0005)
CABoost (our)	0.0103 (± 0.0017)	0.0044 (± 0.0007)	0.0038 (± 0.0007)	0.0042 (± 0.0003)
Simulation error (averaged)	0.0168 (± 0.0630)	0.0063 (± 0.0467)	0.0066 (± 0.0352)	0.0093 (± 0.0424)

5.4 Interpretation of a Prediction

We can explain predictions using model explanation approaches such as SHapley Additive exPlanations (SHAP) [16, 27]. SHAP is a game-theoretic approach for explaining classification and regression models. The algorithm perturbs subsets of input features to take into account the interactions and redundancies between them. The explanation model can then be visualized, showing how the feature values of an instance impact a prediction.

An example of such an explanation is shown in Fig. 3. We can see that both HITS centralities do not impact the explanation much. We can also see that small values of PageRank impact the prediction positively, while small values of Eigenvector and Degree centrality impact the prediction negatively. The above-average out-degree centrality also impacts the model negatively.

6 Discussion and Conclusions

In this paper, we showcase how contemporary graph neural network-based methods can be used for fast estimation of epidemic spreading effect from a given node. We showed that by re-formulating the task as node regression, we can obtain realistic estimates much faster than by performing computationally expensive simulations, even though such simulations are initially used to fine-tune the machine learning models. Further, employment of predictive modeling instead of relying on a single simulation also showed promising results.

We show that while graph neural networks outperform the random baseline and sometimes give us great results, centrality scores and node2vec feature representation coupled with XGBoost mostly outperform them. We also see that machine learning models might overall give a more accurate representation of an epidemic than data gathered from a small number of simulations.

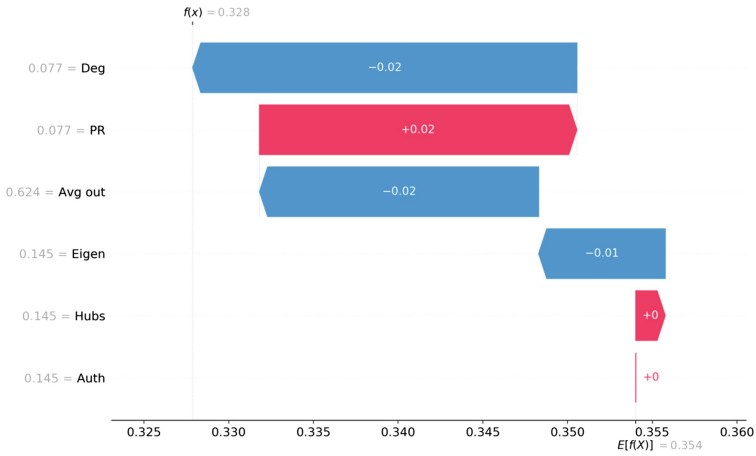


Fig. 3. An example of a model explanation for an instance. Blue arrows indicate how much the prediction is lowered by some feature value, while the red ones indicate how much it is raised. Prediction starts at models expected value 0.354 and finishes at 0.328. Features and their values are shown on the left. The visualization shows for example that the prediction dropped from 0.350 to 0.328 because of the low value of degree centrality.

An obvious limitation of the proposed task is that the spreading is probabilistic and even the best classifiers might make significant errors. Similarly when observing prediction results of the maximum number of infected nodes one must be careful since we predict an average outcome from some node and not the true maximum. This gives us the ability to predict which nodes are most “dangerous” as patient zero. When trying to predict an outcome of an epidemic that has already spread, one must adjust data accordingly and get rid of simulations where epidemics have not spread.

In further work, we plan to research different centralities and algorithms to better describe network structure and achieve more accurate results. Another aspect of our interest is how the proposed method scales and how well it works on different types of networks. We also plan to further research the ability to solve such tasks by using unsupervised algorithms.

Acknowledgments. The work of the last author (BŠ) was funded by the national research agency (ARRS)’s grant for junior researchers. The work of other authors was supported by the Slovenian Research Agency (ARRS) core research program P2-0103 and P6-0411, and research projects J7-7303, L7-8269, and N2-0078 (financed under the ERC Complementary Scheme).

References

1. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016, pp. 785–794, Association for Computing Machinery, New York (2016)
2. Dong, S., Fan, F.-H., Huang, Y.-C.: Studies on the population dynamics of a rumor-spreading model in online social networks. *Phys. A* **492**, 10–20 (2018)
3. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds (2019)
4. Granovetter, M.: Threshold models of collective behavior. *Am. J. Sociol.* **83**(6), 1420–1443 (1978)
5. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864 (2016)
6. Guille, A., Hacid, H., Favre, C., Zighed, D.A.: Information diffusion in online social networks: a survey. *SIGMOD Rec.* **42**(2), 17–28 (2013)
7. Hamsterster. Hamsterster social network. <http://www.hamsterster.com>
8. Kacem, A., Lallemand, C., Giraud, N., Mense, M., De Gennaro, M., Pizzo, Y., Loraud, J.-C., Boulet, P., Porterie, B.: A small-world network model for the simulation of fire spread onboard naval vessels. *Fire Saf. J.* **91**, 441–450 (2017)
9. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2003, pp. 137–146, Association for Computing Machinery, New York (2003)
10. Kermack, W.O., McKendrick, A.G., Walker, G.T.: A contribution to the mathematical theory of epidemics. *Proc. Roy. Soc. Lond. Ser. A, Containing Pap. Math. Phys. Char.* **115**(772), 700–721 (1927)
11. Kesarev, S., Severiukhina, O., Bochenina, K.: Parallel simulation of community-wide information spreading in online social networks. In: Russian Supercomputing Days, pp. 136–148. Springer (2018)
12. Kingma, D.P., Ba, J.: Adam: a Method for Stochastic Optimization. CoRR, abs/1412.6980 (2015)
13. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR) (2017)
14. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. *J. ACM* **46**(5), 604–632 (1999)
15. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Signed networks in social media. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1361–1370. ACM (2010)
16. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 30, pp. 4765–4774. Curran Associates Inc. (2017)
17. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: *Advances in Neural Information Processing Systems*, pp. 4765–4774 (2017)
18. Massa, P., Salvetti, M., Tomasoni, D.: Bowling alone and trust decline in social network sites. In: Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2009. DASC 2009, pp. 658–663. IEEE (2009)

19. Nowzari, C., Preciado, V.M., Pappas, G.J.: Analysis and control of epidemics: a survey of spreading processes on complex networks. *IEEE Control Syst. Mag.* **36**(1), 26–46 (2016)
20. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the Web. In: *WWW 1999* (1999)
21. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: Online learning of social representations. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014*, pp. 701–710. ACM, New York (2014)
22. Rodrigues, F.A.: Network centrality: an introduction. In: *A Mathematical Modeling Approach from Nonlinear Dynamics to Complex Systems*, p. 177 (2019)
23. Rossetti, G., Milli, L., Rinzivillo, S., Sîrbu, A., Pedreschi, D., Giannotti, F.: NDlib: a python library to model and analyze diffusion processes over complex networks. *Int. J. Data Sci. Anal.* **5**(1), 61–79 (2018)
24. Rossi, R.A., Ahmed, N.K.: The network data repository with interactive graph analytics and visualization. In: *AAAI* (2015)
25. Rozemberczki, B., Davies, R., Sarkar, R., Sutton, C.: GEMSEC: graph embedding with self clustering. In: *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2019*, pp. 65–72. ACM (2019)
26. Škrlić, B., Lavrač, N., Kralj, J.: Symbolic graph embedding using frequent pattern mining. In: Novak, P.K., Šmuc, T., Džeroski, S. (eds.) *Discovery Science*, pp. 261–275. Springer, Cham (2019)
27. Štrumbelj, E., Kononenko, I.: Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.* **41**(3), 647–665 (2014)
28. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: *International Conference on Learning Representations* (2018)
29. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* (2020). <https://doi.org/10.1109/TNNLS.2020.2978386>
30. Xiaojin, Z., Zoubin, G.: Learning from labeled and unlabeled data with label propagation. Technical report CMU-CALD-02–107, Carnegie Mellon University (2002)
31. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: *International Conference on Learning Representations* (2019)



Learning Vaccine Allocation from Simulations

Gerrit Großmann^(✉), Michael Backenköhler, Jonas Klesen, and Verena Wolf

Saarland Informatics Campus, Saarland University, 66123 Saarbrücken, Germany
{gerrit.grossmann,mosi}@unisaarland.de

Abstract. We address the problem of reducing the spread of an epidemic over a contact network by vaccinating a limited number of nodes that represent individuals or agents.

We propose a **Simulation-based** vaccine allocation method (**Simba**), a combination of (i) numerous repetitions of an efficient Monte-Carlo simulation, (ii) a PageRank-type influence analysis on an empirical *transmission graph* which is learned from the simulations, and (iii) discrete stochastic optimization.

Our method scales very well with the size of the network and is suitable for networks with millions of nodes. Moreover, in contrast to most approaches that are model-agnostic approaches and solely perform graph-analysis on the contact graph, the stochastic simulations explicitly take the exact diffusion dynamics of the epidemic into account. Thereby, we make our vaccination strategy sensitive to the specific clinical and transmission parameters of the epidemic.

Keywords: SIR Model · Vaccination allocation · Networked epidemic spreading · Control of epidemics · Network robustness and resilience

1 Introduction

Networks provide a universal language to represent interacting systems with emerging dynamical patterns. Computationally, every propagation process over a network can be considered an epidemic. Examples include actual pathogens on human contact networks [1], fake-news in online social networks [8, 10], cascading failures in an infrastructure network [11], congestions in a traffic network, malware in computer networks [3], neural activity in a brain network [4], etc.

The problem of vaccine allocation is linked to the control of such a propagation, where limited vaccine resources are available and we aim to reduce the spread as much as possible, that is, lower the number of nodes reached by the epidemic.

Vaccine allocation strategies can help in the design of complex systems to make them more resilient against (cascading) failures. This is particularly relevant regarding infrastructure networks where a “vaccination” might represent the installation of some kind of protective safeguard. Another example is the mitigation of fake-news in online social networks which can be achieved by removing

the accounts of particularly relevant and malicious influencers or by providing warnings and fact-checking. In the context of infectious diseases, vaccination strategies are an appropriate way of setting priorities in vaccine distribution.

As a model for epidemic spreading, we consider the widely used stochastic, continuous-time Susceptible-Infected-Recovered (SIR) model [9]. Specifically, nodes (eventually) become immune after an infection (or die) and do not transmit the pathogen further. However, our framework is easily adaptable to epidemic models with more disease stages, such as COVID-19 models [5]. We consider as input an (undirected, unweighted) contact network with n nodes and a budget k (number of vaccines). The goal is to identify those k (susceptible) nodes which, when vaccinated, reduce the spread of the epidemic the most. We measure this by using the expected number of susceptible nodes in the terminal state, where the epidemic is over and all nodes are either susceptible, recovered, or vaccinated. In most cases, the only numerically feasible way to approximate this number is to perform a large number of stochastic simulations.

Generally speaking, the vaccination allocation problem is computationally difficult. Intuitively, it is often a good decision to vaccinate those nodes with a large number of neighbors (or with a high *centrality* in the network) and those which are close to the initially infected nodes. If possible, it is even better to identify those nodes which lie between the initially infected nodes and many susceptible nodes. If we represent the spreading process by a *transmission tree* (cf. Fig. 1), in which the direct children of a node v correspond to those nodes that were infected by v , the size of a v 's subtree gives the number of multi-hop infections that originated from v . The premise of our work is that the number of multi-hop infections of a node is a good indicator of whether that node is a good vaccination candidate.

Here, we propose **Simba**, (**S**imulation-based vaccine allocation), which is a method that makes use of recent developments in fast simulation of epidemic processes using a rejection-based approach [6]. This allows performing a large number of simulation runs of networks with millions of nodes and edges in the order of minutes on a standard desktop PC. Based on many simulations, **Simba** constructs a *transmission graph*, a generalization of the transmission tree for several simulation runs. By analyzing this graph, we obtain an impact score for every node. Repeated evaluation of the current vaccination strategy and re-computation of the impact scores yields an iterative optimization procedure, whose objective is to maximize the expected number of nodes that remain healthy.

The key methodological novelty of our proposed vaccination strategy is the construction and analysis of an empirical *transmission graph*. It poses a methodological framework to analyze contagion impact on complex networks. Using the transmission graph, our vaccination strategy can take the dynamics of the epidemic into account. The transmission graph has potentially many more use cases in assessing network dynamics, such as influence maximization, controllability of networks, impact/centrality quantification, and flow prediction. We also provide a numerical evaluation and compare **Simba** to several baselines from the literature.

The manuscript is organized as follows: We first provide a literature overview (Sect. 2), then we formalize the problem statement (Sect. 3). In Sect. 4, we introduce and explain our vaccination allocation method. Experimental results are presented in Sect. 5 and a conclusion completes the manuscript in Sect. 6.

2 Related Work

Traditionally, most methods focused on finding nodes for vaccination using a static analysis of the contact network, for instance by looking at the betweenness centrality of nodes [17] or at their degree [15]. Likewise, NetShield tries to minimize the epidemic threshold of the contact graph (i.e., its general ability to support epidemics) [20]. A more advanced method is GraphShield that starts with degree centrality but then takes the flow of information in the contact graph into account [21]. Eventually, researchers focused more on the dynamical aspects for instance by utilizing linear programming [16] or reinforcement learning [22, 23]. For an overview, we refer the reader to [12].

Conceptually most relevant for us is the work of Zhang et al. who propose DAVA [24] and Song et al. who propose NIIP [18]. Both methods are based on a *dominator tree* architecture which tries to capture the direction of the epidemic. DAVA merges all initially infected nodes and analyzes the paths from this node to all other nodes. Nodes that block a large number of paths are suitable vaccination candidates. NIIP focuses on a problem setting where not all vaccination units are distributed at once. Therefore, NIIP extracts a maximum DAG from the contact graph and uses Monte-Carlo simulation to find the best nodes to vaccinate and combines this with a greedy simulation-based approach, the simulation's goal is to determine *when* to distribute a vaccine.

3 Problem Statement

We first formalize the generative epidemic spreading model and the vaccination allocation problem. We remark that our framework can easily deal with all other types of spreading models as well.

3.1 Continuous-Time Networked SIR Model

Network State. Let $G = (V, E)$ be an undirected, unweighted graph with node set $V = \{v_1, \dots, v_n\}$, containing n nodes, and an edge set E and let any $L : V \rightarrow \{\mathbf{S}, \mathbf{I}, \mathbf{R}\}$ be a node labeling that assigns a *node state* to each node (corresponding to susceptible, infected and recovered nodes). We assume that G is connected (i.e., all nodes are reachable from all other nodes) and has no self-loops. Each labeling function corresponds to a joint state (i.e., a superposition of all node states), called *network state*. The network dynamics specify how the network state (i.e., the labeling) changes over time. We use L_{init} to denote the labeling of the initial network state and we use $S_{\text{init}} (I_{\text{init}}, R_{\text{init}})$ to denote those nodes that were susceptible (infected, recovered) in L_{init} .

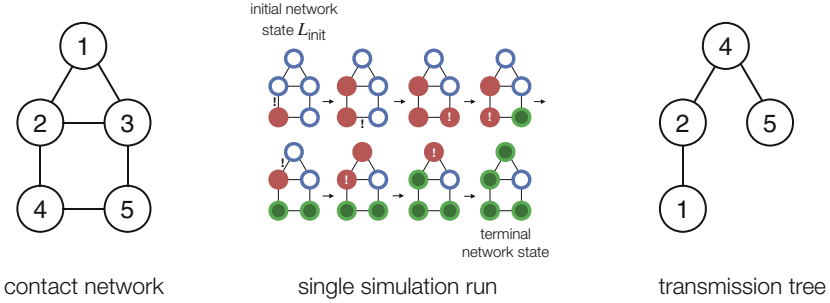


Fig. 1. Example contact network with possible SIR dynamics (S: blue line; I: red, filled; R: green, shaded interior). The firing node/edge is annotated with an exclamation mark. The resulting transmission tree is shown on the right.

Network Dynamics. We also assume that an infection rate constant $\lambda \in \mathbb{R}_{>0}$ and a recovery rate constant $\mu \in \mathbb{R}_{>0}$ are given. W.l.o.g. we typically assume $\mu = 1$ and only vary λ as the spreading dynamics is determined by the fraction $\frac{\lambda}{\mu}$. The network state evolves according to a race condition between the nodes and edges. Generally, all infected nodes can recover at rate μ and each S—I edge can transmit an infection at rate λ , causing the susceptible node to become infected. Consequently, a (computationally naive but statistically correct) simulation run is performed by starting with L_{init} and then, in each simulation step, drawing a firing time for each I-node and for each S—I edge which are exponentially distributed with rates μ and λ , respectively. The event with the shortest firing time “wins” and the corresponding node state (label) is changed accordingly. Repeatedly applying these rules will always lead to a *terminal labeling* or network state where no more actions are possible (all nodes are recovered or susceptible). Given a random simulation run, we use the term *transmission tree* (cf. Fig. 1) to describe a tree where patient zero is the root (if there are more than one infected nodes in the beginning, we merge them) and every node that became infected during the course of the epidemic is connected to the node which infected it. Thus, all nodes in the subtree of a node are called its *children*, i.e. they were directly or indirectly infected by that node.

3.2 Vaccination Allocation Problem

We are given a finite contact network $G = (V, E)$ with corresponding initial labeling L_{init} , a vaccination budget $k \in \mathbb{Z}_{>0}$, as well as the infection and recovery rate constants λ and μ . We want to find a set X of nodes to be vaccinated, where

$$X \subset S_{init} \text{ and } |X| = k . \tag{1}$$

Moreover, for a given $G, L_{init}, k, \lambda, \mu$, we use $F(X)$ to denote the objective function which we define as the expected number of susceptible nodes in the terminal labeling when initially all nodes in X are vaccinated. We define the

Vaccine Allocation Problem as:

Find a set X that maximizes $F(X)$ such that (1) holds.

In practice, we approximate $F(\cdot)$ statistically based on many Monte-Carlo simulation runs. We assume that at least k nodes exist that can be vaccinated and there is at least one infected node in the initial labeling. We model the vaccination by setting $L_{\text{init}}(v) = \mathbf{R}$ for all $v \in X$ at the beginning of the simulation. Note that (assuming the vaccination works perfectly) already recovered, deceased, and vaccinated nodes do not differ from the simulation's point of view.

Complexity. The problem is computationally difficult because there are $\binom{n}{k}$ possibilities to distribute k vaccines to n nodes. The corresponding decision problem is \mathcal{NP} -hard. Specifically, for a given input G , L_{init} , λ , μ , and threshold τ , it is \mathcal{NP} -hard (in n) to decide if a solution X exists s.t. $F(X) > \tau$. It can be shown that for this type of problem, \mathcal{NP} -hardness holds for any propagation model that can mimic an independent cascade (IC) model [24]. We can do this by making μ (λ) arbitrary small (large). Note that this only holds for the SIR model and not necessarily for the generalizations to arbitrary spreading models.

4 Our Method

We first explain the main components of **Simba** (Simulation-based vaccine allocation): the rejection-based simulation method and the construction of the transmission graph with the identification of high-impact nodes based on a ranking analysis.

4.1 Rejection-Based Simulation

For our method, we exploit previous work on fast simulations to efficiently perform a large number of simulations [2, 6]. We propose an algorithm that is statistically equivalent to the generative process description in Sect. 3.1. We perform event-driven simulation using a priority queue. For the initialization, we create one recovery event and one infection event for each infected node and push them into the queue. The firing time is exponentially distributed with rate μ (recovery event) and rate $\lambda \cdot d_i$ (infection from node v_i , d_i being the number of v_i 's neighbors). In each simulation step, we take the first event from the queue. If it is a recovery event, we simply set the corresponding node to state **R**. If it is an infection event, we first check if the corresponding node is still in state **I**, if not, we reject the event and proceed with the next step. If it is, we pick a random neighbor, which will be the target of the infection. We check if the random neighbor is susceptible. If it is, we set the neighbor to **I** and create two events (recovery and infection) for the newly infected neighbor. We also create a new infection event for the source node. Then, we proceed with the next step. The simulation ends when there are no more nodes in state **I**.

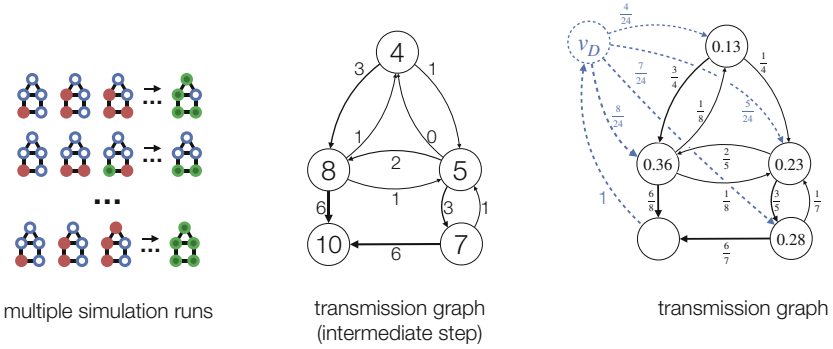


Fig. 2. Schematic illustration of the transmission graph construction based on the same setting as Fig. 1. We consider 10 simulation runs. Center: Contact graph with I_i and $I_{(i,j)}$ as node and edge labels, respectively. Right: Adding the dummy node and normalizing outgoing weights yields a discrete-time Markov chain (DTMC). Nodes in S_{init} are annotated with their impact score based on the equilibrium of the DTMC.

We store the number of susceptible nodes when the simulation ends. Moreover, each time a node gets infected, we store from which (infected) neighbor the infection originated (or all nodes it could have originated from, cf. Sect. 4.2).

4.2 Impact Score Estimation

To estimate the nodes' impacts, we build an empirical *transmission graph* (cf. Fig. 2), an extension of the transmission tree from Fig. 1 to multiple simulation runs. The transmission graph is directed and one can perform a random walk on the graph which (on average) visits nodes with higher impact more often. In the end, we determine the impact of each node in S_{init} by ranking the nodes similar to the idea of the well-known PageRank [13] (i.e., the equilibrium of the corresponding Markov chain).

Given a set of simulated trajectories, let I_i denote the number of trajectories in which node v_i became infected. Furthermore, let $I_{(i,j)}$ denote the number of trajectories in which v_i directly infected v_j . Note that $I_i = \sum_j I_{(j,i)}$.

Transmission Graph. We construct a transmission graph $G_T = (V_T, W_T)$ (with W_T being a weight matrix) as follows: We start with a dummy node v_D as a sink, that is $V_T = V \cup \{v_D\}$ (we can remove unreachable nodes later), and add an edge from each initially infected node with weight one, i.e. for all i :

$$W_T(i, D) = \begin{cases} 1 & \text{if } v_i \in I_{\text{init}}, \\ 0 & \text{otherwise.} \end{cases}$$

Then we add an edge from v_D to each $v_i \in S_{\text{init}}$ with a weight proportional to the estimated probability of that node becoming infected, i.e. for all i :

$$W_T(D, i) = \begin{cases} \frac{I_i}{\sum_{v_j \in S_{\text{init}}} I_j} & \text{if } v_i \in S_{\text{init}}, \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, for all nodes $v_i \neq v_D$ (we consider $0/0$ as 0):

$$W_T(i, j) = \frac{I_{(j,i)}}{I_i}.$$

Note that, by construction, the outgoing weights in G_T are normalized and therefore represent a discrete-time Markov chain. A random walk in the chain will preferably visit nodes of high impact on the epidemic as the transition probabilities are proportional to the estimated infection probabilities. We compute the equilibrium distribution of the corresponding Markov chain using the power iteration method [19]. We call the equilibrium probability of a node normalized over S_{init} its *impact score*. We only consider the impact score for nodes in S_{init} because only those are eligible for vaccination. Note that a transmission graph for a single simulation run is equivalent to the transmission tree where all edges have weight one.

We can make the transmission graph even more accurate. During the simulation, instead of only storing the node that actually transmitted the infection, we store all neighbors that could have potentially been the source of the infection. In our model, each infected neighbor was equally likely to have transmitted the pathogen. It is straightforward to adapt the construction of the transmission graph accordingly even in non-Markovian settings.

4.3 Introducing Simba

We combine the efficient simulations with the transmission graph analysis with an iterative optimization scheme to arrive at **Simba**.

Greedy Initialization. We use C_i to denote the set of vaccinated nodes in iteration i . We start with an empty set, C_0 , of nodes to be vaccinated. Until $|C_i| = k$, we compute the impact score for all nodes $v_i \in S_{\text{init}}$ (assuming nodes in C_i are vaccinated) and add the node with the highest impact to C_i , leading to C_{i+1} .

Optimization. In each optimization step i , we randomly remove one node (with equal probability) from C_i (leading to set B_i) and compute the impact score of all nodes $v_i \in S_{\text{init}} \setminus C_i$ (assuming nodes in B_i are vaccinated). Then we add one of the nodes with the highest impact to B_i (nodes with higher impact are more likely to be chosen), leading to C_{i+1} . We estimate $F(C_i)$ in each iteration step and repeat until some stopping criterion is reached, then we return the set that yielded the highest estimated score.

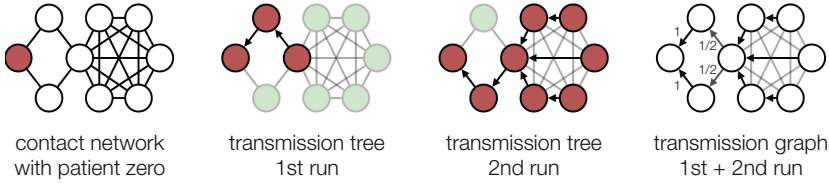


Fig. 3. Assume we want to estimate the impact of the two successor nodes of patient zero based on two simulation runs. Using, for example, the size of their corresponding subtree in the transmission tree leads to misleading results in this case. Specifically, both nodes would be assigned drastically different values. Combining the two runs in a transmission graph yields a more realistic impact score than considering both runs separately. Note that edges point to the origin of the infection and the transmission graph is shown without its dummy node.

4.4 Discussion

Here, we want to address three non-obvious questions: (i) *why build a transmission graph?*, (ii) *what does the graph say about the objective function?*, and (iii) *why is it necessary to consider the dynamics at all?*

Building a Transmission Graph. Using the transmission graph has multiple advantages. Most importantly, transmission trees only mimic a subset of possible infection flows. In contrast, transmission graphs makes it possible to aggregate information over many runs in a principled manner (cf. example in Fig. 3). This way they capture the interplay between connectivity and infection flow more precisely. Moreover, computing the equilibrium of the Markov chain is computationally fast and theoretically well principled. It is also possible to efficiently build the transmission graph on-the-fly during the simulations.

Impact Score and Objective. Note that we handle two different problems. The impact score quantifies the question “*How many nodes became infected as a direct (‘multi-hop’) consequence from each node?*” However, the objective $F(\cdot)$ is concerned with “*How many nodes will (on average) not become infected if a specific (set of) node(s) is vaccinated?*” The latter question is notoriously more difficult to answer. The reason why they differ is that if we vaccinate a node, all of its children in the transmission tree can still become infected via alternative paths. In this sense, the impact score gives an over-approximation on the effect of vaccinating a node regarding the objective. Colloquially, if we vaccinate a node with m children (on average), then the best we can hope for is that these m nodes do not become infected. Hence, our optimization procedure picks nodes depending on their theoretical (and over-approximated) capability or potential to reduce the epidemic spreading.

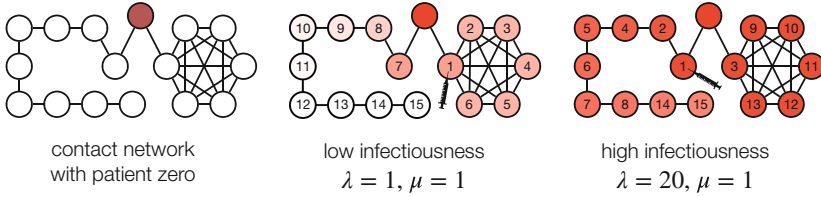


Fig. 4. Assume $k = 1$. The best node to vaccinate depends on the dynamics. If λ is small, the infection will die out on its own in the line graph and it makes more sense to protect the FCC even though it contains fewer nodes. The opacity illustrates a node’s probability to become infected. The nodes are numbered in decreasing order of their impact scores.

Importance of Dynamics. The goal is to vaccinate nodes such that the network becomes less “supportive” of epidemics spreading in it. But then why should the specific dynamics matter? In other words, how can vaccinate a specific node be the right decision for some infection rate constants and the wrong decision for other ones? It is easy to see this in the example in Fig. 4 where we have a single patient zero and a budget of $k = 1$. We can either vaccinate the node to the “right” to protect the fully connected component (FCC) with six nodes or we can vaccinate the node to the “left” to protect the line-graph with nine nodes. If the epidemic is “weak”, it will die out anyway over the line graph, so it makes sense to protect the FCC. In contrast, protecting the line-graph “saves” more nodes if the epidemic is strong enough to conquer the whole graph.

4.5 Generalizations

Our framework can easily be extended to various epidemic-type models. The only necessity is that (i) the model can be simulated (efficiently), (ii) there is a clear objective (e.g., maximize susceptible nodes in terminal states), and (iii) the process represents some contagion phenomena (such that the transmission graph can capture a direction of the information flow). Potential generalizations include models with more disease stages (like SEIR), non-Markovian dynamics (e.g., where the infectiousness of nodes changes over time), weighted and directed networks, as well as temporal or adaptive networks and time-discrete models. Simba can also be adapted to different objectives. For instance, in the SIS model (where infected nodes become susceptible again) the goal is typically to minimize the number of infected nodes in the equilibrium. In that case, our method would identify the nodes that are generally most impactful for the epidemic spreading and not only with regards to a specific initial set of infected nodes. Likewise, we could optimize the timepoints of vaccine distribution [18]. Simba can also be used when the transmission parameters are unknown by using an infection rate constant slightly above the *epidemic threshold* [14].

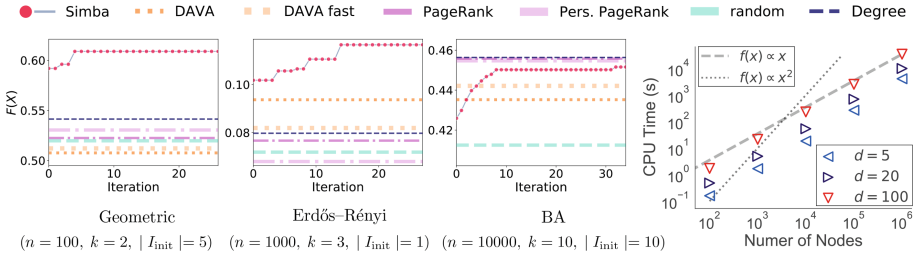


Fig. 5. Left: Optimization of the terminal fraction of expected susceptible nodes $F(X)$ on three sample networks. **Right:** Runtime of 1000 simulations and solution of the corresponding transmission graph based on random d -regular graphs.

5 Experimental Results

We provide an implementation of **Simba** in Rust and Python (for visualization/IO), code will be made available¹. We used synthetic networks following three random graph models (Erdős-Rényi, Geometric, and Barabási-Albert (BA)) with 10^2 , 10^3 , and 10^4 nodes, respectively. The corresponding budgets are $k = 2$, $k = 3$, and $k = 10$. We consider the following baselines: **random** (expected $F(X)$ when random nodes are vaccinated), **DAVA**, and **DAVA-fast** [24], **PageRank**, and **Pers. PageRank** (personalized PageRank) [7, 24], and **Degree** (pick nodes with highest degree). We use 10^3 simulation runs for each construction of the transmission tree. We also analyze the runtime of a complete construction and solution of a transmission graph based on d -regular random graphs (i.e., all nodes have exactly d neighbors) with varying degree d and n . Practically, the runtime is almost linear in n . Theoretically, the number of simulation steps in each run increases linearly. The costs of each simulation step increase sub-linearly. The costs of solving the DTMC also increase linearly. We see that, even though the number of iteration steps is quite small, **Simba** is superior to or (almost) on par with the baselines in the experiments. **Simba** struggles the most with BA graph which is a special case but important as it highlights potential problems. It seems that the general strategy of **Simba** to separate the initially infected from the susceptible nodes does not work better than identifying the nodes which are generally important for the graph’s resilience against epidemics. This is due to the fact that BA graphs typically possess a small subset of nodes that are extremely effective candidates for vaccination regardless of the infection source. Note that **DAVA** also struggles in this case while **Degree** and both **PageRank** methods shine.

6 Conclusions and Future Work

We presented a $novel$ technique to find the most suitable vaccination candidates in a network. Unlike other methods, our approach is based on statistically correct

¹ github.com/gerritgr/Simba.

simulations which are analyzed using the transmission graph. The transmission graph represents the flow of a pathogen in the network as a directed weighted graph. The method is suitable for all epidemic models that can be simulated efficiently.

In the future, we aim to perform different types of information flow analysis on the transmission graph, not only random walks. It remains to be determined which kind of flow analysis is most useful for which type of objective (e.g., vaccination, control, influence maximization). Moreover, we want to extend numerical evaluations to more complex spreading models (e.g., non-Markovian, multi-state ones) and network types (e.g., adaptive networks).

Acknowledgments. This work was partially funded by the DFG project MULTI-MODE.

References

1. Ball, F., Sirl, D., Trapman, P.: Analysis of a stochastic sir epidemic on a random network incorporating household structure. *Math. Biosci.* **224**(2), 53–73 (2010)
2. Cota, W., Ferreira, S.C.: Optimized Gillespie algorithms for the simulation of Markovian epidemic processes on large and heterogeneous networks. *Comput. Phys. Commun.* **219**, 303–312 (2017)
3. Gan, C., Yang, X., Liu, W., Zhu, Q., Zhang, X.: Propagation of computer virus under human intervention: a dynamical model. *Discrete Dynam. Natu. Soc.* **2012** (2012)
4. Goltsev, A., De Abreu, F., Dorogovtsev, S., Mendes, J.: Stochastic cellular automata model of neural networks. *Phys. Rev. E* **81**(6), 061921 (2010)
5. Grossmann, G., Backenköhler, M., Wolf, V.: Importance of interaction structure and stochasticity for epidemic spreading: a covid-19 case study. *ResearchGate* (2020). https://www.researchgate.net/publication/341119247_Importance_of_Interaction_Structure_and_Stochasticity_for_Epidemic_Spreading_A_COVID-19_Case_Study
6. Großmann, G., Wolf, V.: Rejection-based simulation of stochastic spreading processes on complex networks. In: *International Workshop on Hybrid Systems Biology*, pp. 63–79. Springer (2019)
7. Jeh, G., Widom, J.: Scaling personalized web search. In: *Proceedings of the 12th International Conference on World Wide Web*, pp. 271–279 (2003)
8. Khurana, P., Kumar, D.: Sir model for fake news spreading through whatsapp. In: *Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT)*, pp. 26–27 (2018)
9. Kiss, I.Z., Miller, J.C., Simon, P.L., et al.: *Mathematics of Epidemics on Networks*. vol. 598. Springer, Cham (2017)
10. Kitsak, M., Gallos, L.K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H.E., Makse, H.A.: Identification of influential spreaders in complex networks. *Nat. Phys.* **6**(11), 888 (2010)
11. May, R.M., Arinaminpathy, N.: Systemic risk: the dynamics of model banking systems. *J. R. Soc. Interface* **7**(46), 823–838 (2009)
12. Nowzari, C., Preciado, V.M., Pappas, G.J.: Analysis and control of epidemics: a survey of spreading processes on complex networks. *IEEE Control Syst. Mag.* **36**(1), 26–46 (2016)

13. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. Technical report 1999-66, Stanford InfoLab, November 1999. <http://ilpubs.stanford.edu:8090/422/>, previous number = SIDL-WP-1999-0120
14. Prakash, B.A., Chakrabarti, D., Valler, N.C., Faloutsos, M., Faloutsos, C.: Threshold conditions for arbitrary cascade models on arbitrary networks. *Knowl. Inf. Syst.* **33**(3), 549–575 (2012)
15. Prakash, B.A., Tong, H., Valler, N., Faloutsos, M., Faloutsos, C.: Virus propagation on time-varying networks: theory and immunization algorithms. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 99–114. Springer (2010)
16. Sambaturu, P., Vullikanti, A.: Designing robust interventions to control epidemic outbreaks. In: *International Conference on Complex Networks and Their Applications*, pp. 469–480. Springer (2019)
17. Schneider, C.M., Mihaljev, T., Havlin, S., Herrmann, H.J.: Suppressing epidemics with a limited amount of immunization units. *Phys. Rev. E* **84**(6), 061911 (2011)
18. Song, C., Hsu, W., Lee, M.L.: Node immunization over infectious period. In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 831–840 (2015)
19. Stewart, G., Miller, J.: Methods of simultaneous iteration for calculating eigenvectors of matrices. *Top. Numer. Anal.* **II**, 169–185 (1975)
20. Tong, H., Prakash, B.A., Tsourakakis, C., Eliassi-Rad, T., Faloutsos, C., Chau, D.H.: On the vulnerability of large graphs. In: *2010 IEEE International Conference on Data Mining*, pp. 1091–1096. IEEE (2010)
21. Wijayanto, A.W., Murata, T.: Flow-aware vertex protection strategy on large social networks. In: *2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 58–63. IEEE (2017)
22. Wijayanto, A.W., Murata, T.: Learning adaptive graph protection strategy on dynamic networks via reinforcement learning. In: *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pp. 534–539. IEEE (2018)
23. Wijayanto, A.W., Murata, T.: Effective and scalable methods for graph protection strategies against epidemics on dynamic networks. *Appl. Netw. Sci.* **4**(1), 18 (2019)
24. Zhang, Y., Prakash, B.A.: Data-aware vaccine allocation over large networks. *ACM Trans. Knowl. Discov. Data (TKDD)* **10**(2), 1–32 (2015)



Suppressing Epidemic Spreading via Contact Blocking in Temporal Networks

Xunyi Zhao and Huijuan Wang^(✉)

Faculty of Electrical Engineering, Mathematics, and Computer Science,
Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
H.Wang@tudelft.nl

Abstract. In this paper, we aim to effectively suppress the spread of epidemic/information via blocking/removing a given fraction of the contacts in a temporal (time evolving) human contact network. We consider the SI (Susceptible- Infected) spreading process, on a temporal contact network to illustrate our methodology: an infected node infects a susceptible node with a probability β when a contact happens between the two nodes. We address the question: which contacts should be blocked in order to minimize the average prevalence over time. We firstly propose systematically a set of link properties (centrality metrics) based on the aggregated network of a temporal network, that captures the number of contacts between each node pair. Furthermore, we define the probability that a contact $c(i, j, t)$ is removed as a function of the centrality of the corresponding link $l(i, j)$ in the aggregated network as well as the time t of the contact. Each of the centrality metrics proposed can be thus regarded as a contact removal strategy. Empirical results on six temporal contact networks show that the epidemic can be better suppressed if contacts between node pairs that have fewer contacts are more likely to be removed and if contacts happened earlier are likely removed. A strategy tends to perform better when the average number contacts removed per node pair has a lower variance. Strategies that lead to a lower largest eigenvalue of the aggregated network after contact removal do not mitigate the spreading better. This contradicts the finding in static networks, that a network with a small largest eigenvalue tends to be robust against epidemic spreading, illustrating the complexity introduced by the underlying temporal networks.

Keywords: Temporal network · SI spreading · Epidemic mitigation

1 Introduction

Since the outbreak of the Covid-19, most countries have taken mitigation measures to stop or at least reduce the spread. Citizens reduce significantly their transportation and social activities and human contact in general. However, applying the same mitigation measure (e.g. everyone reducing their physical

contact by 10%) to all citizens might not be the most efficient way to stop the virus's spread. The foundational question is which type of social contacts should be blocked in order to slow down the epidemic spreading.

Human contact networks like face-to-face contact networks are in general evolving over time. In such so-called temporal networks, two nodes are connected at a given time step when they have a face-to-face contact. Physical contact networks become available thanks to the development of sensor and communications technologies. In contrast to static networks, where links remain constantly active, the link between a node pair is active (or the two nodes have a contact) only at specific time steps in a temporal network. A temporal network $\mathcal{G} = (\mathcal{N}, \mathcal{C})$ observed within a given time window $[0, T]$ among a set \mathcal{N} of N nodes can be represented by a set of contacts $\mathcal{C} = \{c(i, j, t), t \in [0, T], i, j \in \mathcal{N}\}$, where contact $c(i, j, t)$ occurs between node pair (i, j) at time step t .

In this work, we explore the question which contacts could be removed in order to suppress the epidemic spreading effectively. As a simple start, we consider the Susceptible-Infected (SI) epidemic model, which models information diffusion and epidemic spreading when the spreading is much faster than the recovery. Initially, a seed node is randomly selected and infected at $t = 0$, whereas all the other nodes are susceptible. An infected node infects a susceptible node with a probability β when a contact happens between the two nodes. The prevalence i.e. the percentage of individuals that are infected grows over time. The prevalence over time could be reduced via the removal of contacts. Such reduction in prevalence over time is used to quantify the effect of contact removal. In practice, the temporal contact network at large scale e.g. country level is likely unavailable. We assume that we could obtain the corresponding aggregated network \mathcal{G}_W , where two nodes i and j are connected by a link $l(i, j)$ if the two nodes have at least one contact and the link is associated with a weight representing the number of contacts in between. We aim to design contact removal strategies based on the aggregated network. We propose systematically a set of link centrality metrics or properties based on the aggregated network. Furthermore, we define the probability that a contact $c(i, j, t)$ is removed as a generic function of a centrality metric of link $l(i, j)$ in the aggregated network and the time t of the contact. Each centrality metric thus leads to a different mitigation strategy to select the contacts to block. The average fraction ϕ of contacts to be removed is considered as a control parameter, indicating the mitigation cost. We evaluate the performance of all the strategies that we have proposed in 6 real-world temporal networks. We find that the epidemic prevalence can be better suppressed when contacts between node pairs that have fewer contacts are more likely to be removed, i.e. using the metric one over the number of contacts between a node pair. Removing contacts that happen earlier in time also further enhance the mitigation effect. The number of contacts between a node pair is heterogeneous. It seems that the mitigation effect is better if the average number of contacts removed per node pair varies less. Static network studies have shown that a weighted network tends to be more robust against epidemic spreading with respect to its epidemic threshold if its largest eigenvalue is smaller. The resultant aggregated network after contact removal, however, may have a lower

prevalence if its largest eigenvalue is larger. This implies that the underlying temporal network may lead to new phenomena in epidemic spreading that differ from what we have learned from static networks.

The influence of temporal networks on dynamic processes has been widely investigated [1–3]. Gemmetto et al. have studied the epidemic mitigation via excluding a sub-group of nodes in a temporal network [10]. Link blocking strategies using link centrality metrics to suppress information diffusion has been explored in [4]. The links to block are selected from the aggregated network. When a link is blocked, all contacts associated with the links are all removed. In this work, we investigate more in-depth at contact level, i.e. how to choose the contacts to block when the total number of contacts to block is given. Moreover, the consideration of the time of a contact in contact removal strategies may inspire the decision when a mitigation should be implemented.

2 Methods

We propose firstly a set of link centrality metrics/properties based on the aggregated network \mathcal{G}_W . Furthermore, the probability that a contact is removed is defined step by step as a function of a given centrality metric and the time of the contact, which also ensures that a fraction ϕ of contacts are removed on average. We evaluate the effect of the mitigation strategies via the extent that the prevalence is reduced over time.

2.1 Link Centrality Metrics

An aggregated network \mathcal{G}_W can be constructed based on the temporal network \mathcal{G} observed over time window $[1, T]$. We propose the following link centrality metrics based on the weighted aggregated network:

- *Degree product*: the product of the degrees of the two end nodes of a link, where the nodal degree is defined as the number of neighbors of a node.
- *Strength product*: the product of the strengths of the two end nodes of a link, where the nodal strength is the sum of weights of all the links incident to the node, or equivalently the total number of contacts the node involves in the temporal network.
- *Betweenness*: the number of shortest paths in the unweighted aggregated network that traverse the link between all possibly node pairs.
- *Link weight*: the weight of a link in the aggregated network. It is the same as the number of contacts between the two end nodes in the temporal network.
- *Weighted eigenvector component product*: the product of the principal eigenvector components of the two end nodes, where the principal eigenvector is the eigenvector corresponds to the largest eigenvalue of the weighted aggregated network.
- *Unweighted eigenvector component product*: the product of the principal eigenvector components of the two end nodes, where the principal eigenvector is the eigenvector corresponds to the largest eigenvalue of the unweighted aggregated network.
- *Random*: the metric is set as 1 for all links.

For each metric m_{ij} , we consider $\frac{1}{m_{ij}}$ as an extra centrality metric, except for the *random*. For any centrality metric, the centrality value of every link in the aggregated network is positive. The motivation to consider the reciprocal metrics $\frac{1}{m_{ij}}$ is explained in the design of the removal probabilities of contacts (2).

Link centrality metrics can be correlated [5, 6]. We find that the Spearman rank correlation between any two metrics proposed above is weak, i.e. below 0.2. This implies that each metric captures a specific property that can not be captured by another metric.

2.2 Contact Removal Probability

For a given link centrality metric, we can compute the centrality for m_{ij} for each link $l(i, j)$. We propose the probability p_{ij} that a contact $c(i, j, t)$ between i and j is removed as

$$p_{ij} = m_{ij} \frac{\phi \sum_{ij} w_{ij}}{\sum_{ij} (w_{ij} m_{ij})} \quad (1)$$

where w_{ij} is the weight of link $l(i, j)$ in the aggregated network, and the normalization ensures that on average a fraction ϕ of contacts will be removed. The probability that a contact is removed is assumed to be proportional to the centrality m_{ij} of the corresponding link $l(i, j)$.

We found that some centrality metrics are highly heterogeneous. Hence, it is possible that the removal probability calculated by (1) is larger than 1 for contacts whose associated link $l(i, j)$ has an extremely large centrality m_{ij} . In such cases, the actual fraction of contacts removed can be lower than the expected ϕ , if all contacts with removal probability larger than 1 are removed. Therefore, we set the removal probabilities of those contacts to 1 and re-normalize the removal probability among the rest contacts. This process is repeated until the removal probabilities of all remaining contacts are between 0 and 1, while the actual fraction of contacts removed is the same as expected ϕ .

The probability p_{ij} that a contact between i and j is removed can be defined in a more general way

$$p_{ij}^* = m_{ij}^\alpha \frac{\phi \sum_{ij} w_{ij}}{\sum_{ij} (w_{ij} m_{ij}^\alpha)} \quad (2)$$

The removal probability of a contact between i and j is proportional to a polynomial function of m_{ij} . Our choice in (1) corresponds to the case when $\alpha = 1$. The random strategy, i.e. every contact has the same probability to be removed, corresponds to the case when $\alpha = 0$. The choice of the reciprocal metric $\frac{1}{m_{ij}}$ in (1) is equivalent to the general definition (2) when metric m_{ij} is considered and $\alpha = -1$. Hence, we consider removal probability (1) using the list of centrality metrics proposed and their reciprocals as well as the random strategy, which correspond to the general definition of (2) where $\alpha = 1, -1, 0$, respectively.

Furthermore, we wonder whether removing contacts that happen earlier or introducing the mitigation intervention earlier in time would better reduce the

prevalence. To take the time factor of the contacts into account, we propose the probability $p_{ij}(t)$ that a contact $c(i, j, t)$ between i and j at t is removed as

$$p_{ij}(t) = m_{ij} f(t) \frac{\phi \sum_{ij} w_{ij}}{\sum_{ij} (w_{ij} m_{ij} f(t))} \quad (3)$$

where $f(t)$ implies the preference to block contacts at specific period. The probability that $c(i, j, t)$ is removed is proportional to $m_{ij} \cdot f(t)$.

As a simple start, we consider $f(t) = 4 \cdot 1_{t \leq T/2} + 1_{t > T/2}$, $f(t) = 1_{t \leq T/2} + 4 \cdot 1_{t > T/2}$ and $f(t) = 1$, where the indicator function 1_y is one if the condition y is true, and otherwise it is 0. These three functions correspond to the preference to block contacts happening in $[1, T/2]$, in $(T/2, T]$ and no preference over the time of the contacts, respectively.

2.3 Datasets

We consider six real-world temporal physical contact networks, measured in three scenarios:

- HighSchool11&12 [7] capture the physical contacts between students in a high school in Marseilles, France. The two datasets consider two different groups of students.
- Workplace13&15 [8] are the temporal networks of contacts between individuals measured in an office building in France. Different groups of individuals are considered in the two datasets respectively.
- MIT1&2 [9] contain human contact data among students of the Massachusetts Institute of Technology. In order to keep the duration of the observation time window relatively comparable with the other networks, we randomly select two one-week periods as two temporal networks.

All networks are considered as undirected. Some basic properties of the networks are shown in Table 1.

Table 1. Basic properties of the temporal networks: the number of nodes, links and contacts. The duration is the duration of the observation time window $[1, T]$ measured in days, thus T times the duration per discrete time step.

Datasets	Nodes	Links	Contacts	Duration
HighSchool11	126	1709	28561	3.15
HighSchool12	180	2220	45047	8.44
WorkPlace13	92	755	9827	11.43
WorkPlace15	217	4274	78249	11.50
MIT1	74	355	29107	6.99
MIT2	45	200	22714	6.99

2.4 Simulation

We consider as an example the infection probability $\beta = 0.01$, the probability that a susceptible gets infected by an infected node when they have a contact. This infection probability leads to a prevalence around the order of 10% by the end of the time window in each temporal network.

For each centrality metric and each temporal network, we select each node in the network as a possible seed node. For each seed node, we iterate the following for five times. In each iteration, the fraction ϕ of contacts to be removed are selected according to the centrality metric thus the probability (1) using the given link centrality metric; The SI process starting from the given seed is performed on the temporal network where the selected contacts are removed; the prevalence ρ over time is recorded. For each network and centrality metric, we could obtain the prevalence at any time as the average over all possible seed nodes and the five iterations for each seed node. The fraction ϕ of contacts to be removed is a control parameter and $\phi = 10\%$ and $\phi = 30\%$ have been considered.

Simulations are performed in the same way when the time factor $f(t)$ are taken into account via the removal probability of a contact defined in (3).

3 Results

First of all, we evaluate the performance of all strategies as defined in (1) where the time of a contact has no influence on its probability of being blocked.

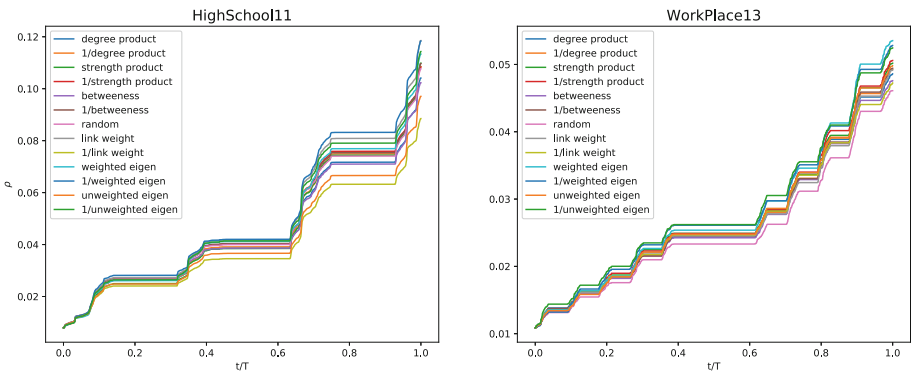


Fig. 1. The prevalence ρ over time, when $\phi = 10\%$ of the contacts are removed using each centrality metric according to (1) in two temporal networks.

Figure 1 illustrates how the prevalence ρ grows over time when each contact blocking strategy is performed in two networks and 10% contacts are removed.

Table 2. The prevalence $E[\rho]$ averaged over time when $\phi = 10\%$ of the contacts are removed from each temporal network using removal probability (1) based on each centrality metric. The best performance in each network is marked in bold.

Metrics	HighSchool11	WorkPlace13	MIT1	HighSchool12	WorkPlace15	MIT2
Degree product	0.0444	0.0264	0.1128	0.0438	0.1008	0.1903
1/degree product	0.0457	0.0264	0.1050	0.0455	0.1051	0.1765
Strength product	0.0450	0.0269	0.1154	0.0445	0.1082	0.1998
1/strength product	0.0465	0.0270	0.0968	0.0416	0.0969	0.1605
Betweenness	0.0442	0.0261	0.0990	0.0392	0.1042	0.1961
1/betweenness	0.0465	0.0264	0.1169	0.0447	0.1035	0.1929
Random	0.0459	0.0250	0.1106	0.0430	0.1010	0.1909
Link weight	0.0488	0.0263	0.1227	0.0483	0.1131	0.1946
1/link weight	0.0396	0.0263	0.0922	0.0355	0.0836	0.1689
Weighted eigen	0.0470	0.0277	0.1180	0.0453	0.1039	0.2071
1/weighted eigen	0.0499	0.0280	0.0976	0.0441	0.1029	0.1661
Unweighted eigen	0.0417	0.0267	0.1126	0.0435	0.0988	0.1981
1/unweighted eigen	0.0478	0.0283	0.1080	0.0430	0.1099	0.1861

The performance of the strategies in each network can be also compared via the average prevalence $E[\rho]$ over the whole time window, as shown in Table 2 and 3, where $\phi = 10\%$ and $\phi = 30\%$ percent of contacts are removed respectively. The 1/link weight performs the best in all networks except for MIT2 and/or WorkPlace13. These observations imply that it is more effective to suppress the epidemic by removing contacts between node pairs that have few contacts.

For any node pair (i, j) , the average number of contacts removed between i and j is $p_{ij}w_{ij}$. For strategy 1/link weight, the average number of contacts removed is the same for all node pairs or for all links in the aggregated network¹. We wonder whether a more similar number of contacts removed per node pair leads to a better mitigation effect. Hence, we derive further the variance $Var[p_{ij}w_{ij}]$ for each strategy in each network. Figure 2(a) shows the scatter plot of the average prevalence $E[\rho]$ versus $\sqrt{Var[p_{ij}w_{ij}]}$. In each network, a strategy tends to perform better i.e. leads to a low $E[\rho]$ if the $Var[p_{ij}w_{ij}]$ is small.

In the studies of the Susceptible-Infected-Susceptible SIS epidemic spreading model on a static weighted network, the largest eigenvalue of the weighted network has been shown to suggest the robustness of the network against epidemic [11–14]. The infection rate between two individuals is assumed in the SIS model to be proportional to the infection rate of the epidemic multiply by

¹ For strategy 1/link weight, the actual average number of contacts removed per node pair in the simulation may differ slightly among the links, because when the removal probability p_{ij} derived from (1) is larger than one, we set $p_{ij} = 1$, and re-normalize the removal probabilities of the rest links so that a fraction ϕ of contacts are removed as expected.

Table 3. The prevalence $E[\rho]$ averaged over time when $\phi = 30\%$ of the contacts are removed from each temporal network using removal probability (1) based on each centrality metric.

Metrics	HighSchool11	WorkPlace13	MIT1	HighSchool12	WorkPlace15	MIT2
Degree product	0.0266	0.0212	0.0929	0.0273	0.0521	0.1732
1/degree product	0.0375	0.0219	0.0847	0.0322	0.0712	0.1447
Strength product	0.0363	0.0231	0.1032	0.0321	0.0656	0.1773
1/strength product	0.0313	0.0225	0.0673	0.0302	0.0645	0.1087
Betweenness	0.0313	0.0227	0.0796	0.0284	0.0574	0.1495
1/betweenness	0.0353	0.0231	0.1020	0.0318	0.0653	0.1630
Random	0.0320	0.0216	0.0881	0.0298	0.0634	0.1717
Link weight	0.0431	0.0240	0.1008	0.0398	0.0874	0.1785
1/link weight	0.0210	0.0202	0.0572	0.0191	0.0391	0.1170
Weighted eigen	0.0343	0.0242	0.1016	0.0337	0.0673	0.1782
1/weighted eigen	0.0395	0.0227	0.0668	0.0340	0.0709	0.1030
Unweighted eigen	0.0264	0.0218	0.0950	0.0290	0.0557	0.1645
1/unweighted eigen	0.0383	0.0215	0.0826	0.0325	0.0708	0.1390

the link weight, i.e. the contact frequency. In this case, the epidemic threshold $\tau_c \sim \frac{1}{\lambda_1(W)}$, where matrix W with its element w_{ij} captures the weights of all links in the aggregated network. When the effective infection rate, i.e. infection rate normalized by the recovery rate of the epidemic, is above the threshold, a none-zero fraction of the population gets infected in the meta-stable state, whereas below the threshold, the epidemic dies out in the meta-stable state. A static weighted network with a small largest eigenvalue tends to be more robust against epidemic. We explore further the largest eigenvalue $\lambda_1(W^*)$ of the resultant aggregated network after contact removal whose weighted adjacency matrix is W^* . Would a strategy that leads to a smaller $\lambda_1(W^*)$ be more effective in suppress the prevalence according to the findings of SIS model on static networks? The scatter plot in Fig. 2(b) of the average prevalence $E[\rho]$ versus $\lambda_1(W^*)$ shows the contrary: the prevalence tends to be low when the resultant network has a large largest eigenvalue. This inconsistency can be possibly introduced by the following. Removing many contacts from few links whose end nodes have a high strength may better reduce the largest eigenvalue. This is less effective in mitigation an SI spreading process where each link can transmit the epidemic maximally once dependent also on the time ordering of contacts. It can be, however, effective to mitigate an SIS epidemic where such links could transmit the epidemic frequently.

Finally, we take the time of a contact into account when selecting the contacts to remove via the contact removal probability $p_{ij}(t)$ defined in (3). When $f(t) = 1_{t \leq T/2} + 4 \cdot 1_{t > T/2}$, contacts happening late i.e. $t > T/2$ in time are more likely to be removed. When $f(t) = 4 \cdot 1_{t \leq T/2} + 1_{t > T/2}$, contacts happening early i.e. $t < T/2$ are 4 times more likely to be removed compared to contacts happening late $t > T/2$. Comparing Table 3, 4 and 5, where the contact removal is independent

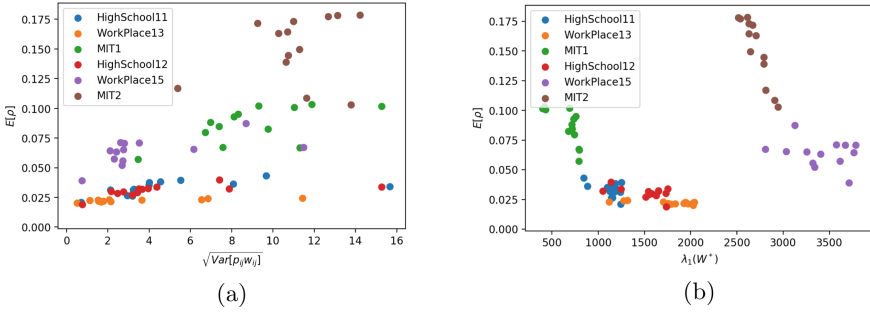


Fig. 2. (a) Scatter plot of the average prevalence $E[\rho]$ versus the standard deviation $\sqrt{\text{Var}[\rho_{ij} w_{ij}]}$ of the average number of contacts removed from a node pair. (b) Scatter plot of the average prevalence $E[\rho]$ versus the largest eigenvalue $\lambda_1(W^*)$ of the resultant aggregated network after the contact removal. A fraction $\phi = 30\%$ of contacts are removed.

of the time of a contact, favors the removal of late and early contacts respectively, we find that the suppressing effect is better when early contacts are more likely to be removed. Furthermore, the metric 1/link weight tends to perform the best independent of the choice of $f(t)$. Hence, the mitigation effect tends to be better if contacts between node pairs that have few contacts and earlier contacts are more likely to be removed. Node pairs with few contacts are usually referred as weak social ties. Removing the contacts along weak social ties seems an effective and likely socially feasible mitigation strategy.

Table 4. The prevalence $E[\rho]$ averaged over time when $\phi = 30\%$ of the contacts are removed from each temporal network using contact removal probability (3) and $f(t) = 1_{t \leq T/2} + 4 \cdot 1_{t > T/2}$ based on each centrality metric. Contacts happening late i.e. $t > T/2$ in time are more likely to be removed.

Metrics	HighSchool11	WorkPlace13	MIT1	HighSchool12	WorkPlace15	MIT2
Degree product	0.0341	0.0241	0.1027	0.0349	0.0717	0.1878
1/degree product	0.0418	0.0235	0.0981	0.0359	0.0807	0.1543
Strength product	0.0390	0.0236	0.1093	0.0352	0.0717	0.1920
1/strength product	0.0358	0.0234	0.0734	0.0320	0.0741	0.1315
Betweenness	0.0342	0.0236	0.0899	0.0321	0.0715	0.1663
1/betweenness	0.0406	0.0242	0.1094	0.0385	0.0828	0.1809
Random	0.0385	0.0260	0.1033	0.0353	0.0751	0.1815
Link weight	0.0443	0.0252	0.1156	0.0417	0.0932	0.1992
1/link weight	0.0245	0.0217	0.0669	0.0212	0.0453	0.1362
Weighted eigen	0.0366	0.0236	0.1144	0.0342	0.0714	0.1863
1/weighted eigen	0.0436	0.0250	0.0690	0.0345	0.0745	0.1196
Unweighted eigen	0.0327	0.0234	0.1085	0.0360	0.0681	0.1804
1/unweighted eigen	0.0444	0.0255	0.0936	0.0366	0.0843	0.1519

Table 5. The prevalence $E[\rho]$ averaged over time when $\phi = 30\%$ of the contacts are removed from each temporal network using contact removal probability (3) and $f(t) = 4 \cdot 1_{t \leq T/2} + 1_{t > T/2}$ based on each centrality metric. Contacts happening early i.e. $t < T/2$ in time are more likely to be removed.

Metrics	HighSchool11	WorkPlace13	MIT1	HighSchool12	WorkPlace15	MIT2
Degree product	0.0244	0.0191	0.0696	0.0251	0.0460	0.1343
1/degree product	0.0333	0.0208	0.0765	0.0277	0.0585	0.1195
Strength product	0.0318	0.0210	0.0803	0.0283	0.0558	0.1436
1/strength product	0.0273	0.0206	0.0612	0.0286	0.0539	0.1047
Betweenness	0.0269	0.0194	0.0657	0.0242	0.0463	0.1244
1/betweenness	0.0275	0.0201	0.0910	0.0282	0.0538	0.1343
Random	0.0265	0.0196	0.0766	0.0267	0.0528	0.1345
Link weight	0.0417	0.0233	0.0867	0.0345	0.0738	0.1546
1/link weight	0.0193	0.0185	0.0480	0.0179	0.0322	0.1074
Weighted eigen	0.0318	0.0242	0.0940	0.0332	0.0611	0.1474
1/weighted eigen	0.0419	0.0221	0.0653	0.0328	0.0613	0.0983
Unweighted eigen	0.0223	0.0196	0.0756	0.0238	0.0456	0.1421
1/unweighted eigen	0.0366	0.0218	0.0724	0.0286	0.0612	0.1226

4 Conclusion and Discussion

In this work, we have introduced the methodology of suppressing the epidemic spreading via removing a given fraction of contacts in a temporal network. The choice of the contacts to remove is designed in a generic and probabilistic way. The probability that a contact $c(i, j, t)$ is removed is a function of the centrality or property of the corresponding link $l(i, j)$ in the aggregated network as well as the time t of the contact. A large number of relatively independent link centrality metrics have been considered. We find that removing the contacts between the node pairs that have few contacts and removing contacts in an earlier phase tend to suppress the prevalence more. This implies that the removal of contacts along weak social ties in an early phase tends reduce the prevalence more effectively. On the other hand, removing the large number of contacts of few node pairs is likely too costly to be effective.

To illustrate the methodology, we have confined ourselves to the SI spreading model, limited number of real-world networks and limited choices of the parameters. Our methods may inspire further studies beyond the limited scenarios that we have considered. Our mitigation method is based on the aggregated network over the whole time window $[1, T]$, when the mitigation is supposed to be carried out. It is interesting to explore whether we can estimate this aggregated network based on the observation of the aggregated network in the past. The performance of the mitigation strategies may depend on the properties of the underlying temporal networks. A fundamental question is which temporal network properties favor which mitigation strategies. This requires the expertise in the modeling of temporal networks and temporal network randomization. The effect of mitigation strategies depends as well on the relative spreading probability/rate. When

an epidemic spreads extremely fast, e.g. all nodes have already been infected before $T/2$, the aggregated network information is likely not ideal to determine the contact removal probabilities, though this scenario is less realistic.

Acknowledgement. The authors wish to thank TU Delft COVID-19 Response Fund.

References

1. Holme, P., Saramäki, J.: Temporal networks. *Phys. Rep.* **519**, 97–125 (2012)
2. Holme, P.: Modern temporal network theory: a colloquium. *Eur. Phys. J. B* **88**, 234 (2015)
3. Zhan, X., Hanjalic, A., Wang, H.: Information diffusion backbones in temporal networks. *Sci. Rep.* **9**, 6798 (2019)
4. Zhan, X., Hanjalic, A., Wang, H.: Suppressing information diffusion via link blocking in temporal networks. In: Cherifi, H., Gaito, S., Mendes, J.F., Moro, E., Rocha, L.M. (eds.) *Complex Networks and Their Applications VIII*, vol. 881, pp. 448–458. Springer, Heidelberg (2020)
5. Li, C., Li, Q., Van Mieghem, P., Stanley, H.E., Wang, H.: Correlation between centrality metrics and their application to the opinion model. *Eur. Phys. J. B* **88**, 65 (2015)
6. Li, C., Wang, H., de Haan, W., Stam, C.J., Mieghem, P.V.: The correlation of metrics in complex networks with applications in functional brain networks. *J. Stat. Mech.* **2011**, P11018 (2011)
7. Fournet, J., Barrat, A.: Contact patterns among high school students. *PLoS ONE* **9**, e107878 (2014)
8. Enois, M., Vestergaard, C., Fournet, J., Panisson, A., Bonmarin, I., Barrat, A.: Data on face-to-face contacts in an office building suggest a low-cost vaccination strategy based on community linkers. *Netw. Sci.* **3**, 326–347 (2015)
9. Reality mining network dataset–KONECT. <https://konect.uni-koblenz.de/networks/mit>
10. Gemmetto, V., Barrat, A., Cattuto, C.: Mitigation of infectious disease at school: targeted class closure vs school closure. *BMC Infect. Dis.* **14**, 695 (2014)
11. Van Mieghem, P., Omic, J., Kooij, R.: Virus spread in networks. *IEEE/ACM Trans. Netw.* **17**, 1–14 (2009)
12. Wang, H., Li, Q., D’Agostino, G., Havlin, S., Stanley, H.E., Van Mieghem, P.: Effect of the interconnected network structure on the epidemic threshold. *Phys. Rev. E* **88**, 022801 (2013)
13. Ottaviano, S., De Pellegrini, F., Bonaccorsi, S., Mugnolo, D., Van Mieghem, P.: Community networks with equitable partitions. In: Altman, E., Avrachenkov, K., De Pellegrini, F., El-Azouzi, R., Wang, H. (eds.) *Multilevel Strategic Interaction Game Models for Complex Networks*, pp. 111–129. Springer, Heidelberg (2019)
14. Qu, B., Wang, H.: SIS epidemic spreading with heterogeneous infection rates. *IEEE Trans. Netw. Sci. Eng.* **4**, 177–186 (2017)



Blocking the Propagation of Two Simultaneous Contagions over Networks

Henry L. Carscadden¹, Chris J. Kuhlman¹, Madhav V. Marathe¹,
S. S. Ravi^{1,2}(✉), and Daniel J. Rosenkrantz^{1,2}

¹ University of Virginia, Charlottesville, USA
{h1c5v,cjk8gx,marathe}@virginia.edu

² University at Albany – SUNY, Albany, USA
ssravi0@gmail.com, drosenkrantz@gmail.com

Abstract. We consider the simultaneous propagation of two contagions over a social network. We assume a threshold model for the propagation of the two contagions and use the formal framework of discrete dynamical systems. In particular, we study an optimization problem where the goal is to minimize the total number of infected nodes subject to a budget constraint on the total number of nodes that can be vaccinated. While this problem has been considered in the literature for a single contagion, our work considers the simultaneous propagation of two contagions. Since the optimization problem is **NP**-hard, we develop a heuristic based on a generalization of the set cover problem. Using experiments on three real-world networks, we compare the performance of the heuristic with some baseline methods.

1 Introduction

Contagion models have been used to explain a host of observed phenomena in human populations (e.g., the spread of diseases, fads, opinions, information, actions such as joining a group) [8, 16, 19]. In this paper, we treat contagions as undesirable entities (such as infectious diseases) propagating through a network. Network models of contagion propagation capture complex patterns of interaction missed by models that assume homogeneous mixing. These interactions present interesting combinatorial optimization problems such as seed selection and contagion blocking. Our focus in this paper is on blocking. Previous work on blocking focuses on the case where only a single contagion is propagating through a network (see, e.g., [5, 11] and the references cited therein). We seek to extend prior work from the single contagion setting to the multiple contagion setting. To understand the landscape of the area, we consider two independent contagions propagating under the threshold model [9]. Under this model, an individual (i.e., node in a social network) gets infected because it has at least a sufficient number (called the **threshold**) of infected neighbors. In addition to disease propagation, threshold models [4, 9, 17, 21] have also been used to capture other social contagions (such as information, opinion and fads). In this paper, we

consider disease propagation and use vaccinating nodes as the blocking strategy. The goal is to reduce the number of newly infected nodes under a budget on the number of nodes that can be vaccinated. Following [11], we use the synchronous dynamical system (SyDS) as the formal model for contagion propagation; see Sect. 2.

Summary of Results: We discuss a general threshold-based model for the simultaneous propagation of two contagions through a network. As this general model (which requires the specification of five threshold values for each node) is somewhat complex, we consider a simplified model that uses only two threshold values for each node. Using that model, we formulate the problem of minimizing the number of new infections in a network by vaccinating some nodes. In practice, there is a budget constraint on the number of vaccinations. We observe that the resulting budget-constrained optimization problem is computationally intractable using a known result for the case of a single contagion [11]. Therefore, we develop an efficient heuristic algorithm called MCICH for the problem. This heuristic is based on a generalized version of the Minimum Set Cover (MSC) problem [7]. Through computational experiments, we compare the performance of MCICH with two baseline methods using three real-world social networks. Our results indicate that MCICH is able to block the two contagions effectively even with a small vaccination budget, and performs far better than the other two methods.

Related Work: Reference [11] treats the single contagion blocking problem under the threshold model. The goal is again to minimize the number of new infections subject to a budget on the number of nodes that can be vaccinated. It is shown that if the budget cannot be violated, even obtaining an approximation algorithm with any provable performance guarantee is **NP**-hard. Two efficient heuristics for the problem are introduced and their performance is evaluated on several social networks. Although single contagion epidemic models have been studied for years, study of the multiple contagion context is newer. For example, conditions for the coexistence of two contagions in compartmental models are explored in [3]. A number of references (see e.g., [10, 14, 15] and the references cited therein) have considered the propagation of competing contagions (where infection by one contagion prevents or reduces the likelihood of infection by another), and cooperating contagions (where infection by one contagion makes it easier to get infected by another contagion). While our work uses the deterministic threshold model, reference [18] discusses a general framework for a probabilistic multiple-contagion model, namely the Susceptible-Infected-Recovered (SIR) model.

2 Definitions and Analytical Results

Model Description: We use the **synchronous dynamical system** (SyDS) model studied in the literature (see e.g., [2]). A (SyDS) \mathbb{S} over a domain \mathbb{B} is specified as a pair $\mathbb{S} = (G, \mathbb{F})$, where (a) $G(V, E)$, an undirected graph with

Table 1. Possible states for each node

State	Interpretation
0	Not infected by either \mathbb{C}_1 or \mathbb{C}_2
1	Infected by \mathbb{C}_1 only
2	Infected by \mathbb{C}_2 only
3	Infected by both \mathbb{C}_1 and \mathbb{C}_2

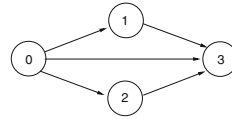


Fig. 1. Possible state transitions for each node

$|V| = n$, represents the underlying graph of the SyDS, with node set V and edge set E , and (b) $\mathbb{F} = \{f_1, f_2, \dots, f_n\}$ is a collection of functions in the system, with f_i denoting the **local function** associated with node v_i , $1 \leq i \leq n$. Each node of G has a state value from \mathbb{B} . Each function f_i specifies the local interaction between node v_i and its neighbors in G . The inputs to function f_i are the state of v_i and those of the neighbors of v_i in G ; function f_i maps each combination of inputs to a value in \mathbb{B} . This value becomes the next state of node v_i . It is assumed that each local function can be computed efficiently.

For a single contagion, the domain \mathbb{B} is usually chosen as $\{0,1\}$, with 0 and 1 representing that a node is uninfected and infected respectively. Since we have two contagions (denoted by \mathbb{C}_1 and \mathbb{C}_2) propagating through the underlying network, we have four possible states for each node, denoted by 0, 1, 2 and 3; thus, $\mathbb{B} = \{0, 1, 2, 3\}$.

The interpretation of these state values is shown in Table 1. An easy way to think of these states is to consider the 2-bit binary expansion of the state values 0 through 3. The least (most) significant bit indicates whether the node has been infected by \mathbb{C}_1 (\mathbb{C}_2).

We assume that the system is **progressive** with respect to each of the contagions [6]; that is, once a node is infected by a contagion, it remains infected by that contagion. Using this assumption, Fig. 1 shows the possible state transitions for each node.

State Transition Rules: Each node v is associated with a **local transition function** f_v that determines the next state of v given its current state and the states of the neighbors of v . Such a function may be deterministic or stochastic (as in SIR systems). Here, we will consider a simple class of deterministic functions called **threshold functions**.

A General Form of Local Functions: We first discuss a very general (but somewhat complex) form of local functions for the propagation of two contagions in a network and then present a simpler form that will be used in the paper. In the general form, for each node v and each of the five possible state transition x to y (shown in Fig. 1), there is a threshold value $\theta(v, x, y)$. Let $N(v, j)$ denote the number of neighbors of v in state j , $0 \leq j \leq 3$. (If the state of node v is j , then v is included in the count $N(v, j)$.) For any node v , the rules for each possible state transition which collectively specify the local function f_v are shown in Table 2.

Table 2. Transition rules to specify the local function f_v

Transition	Condition
$0 \rightarrow 1$	$(N(v, 1) + N(v, 3) \geq \theta(v, 0, 1))$ and $(N(v, 2) + N(v, 3) < \theta(v, 0, 2))$
$0 \rightarrow 2$	$(N(v, 1) + N(v, 3) < \theta(v, 0, 1))$ and $(N(v, 2) + N(v, 3) \geq \theta(v, 0, 2))$
$0 \rightarrow 3$	$(N(v, 1) + N(v, 3) \geq \theta(v, 0, 1))$ and $(N(v, 2) + N(v, 3) \geq \theta(v, 0, 2))$
$1 \rightarrow 3$	$N(v, 2) + N(v, 3) \geq \theta(v, 1, 3)$
$2 \rightarrow 3$	$N(v, 1) + N(v, 3) \geq \theta(v, 2, 3)$

We briefly explain two of the state transition conditions shown in Table 2. The conditions for other state transitions are similar. Consider the condition for the “ $0 \rightarrow 1$ ” transition. For this transition to occur at a node v , the number of neighbors of v in state 1 or state 3 must be at least $\theta(v, 0, 1)$ (i.e., $(N(v, 1) + N(v, 3) \geq \theta(v, 0, 1))$) and the number of neighbors of v in state 2 or state 3 must be *less than* $\theta(v, 0, 2)$ (i.e., $(N(v, 2) + N(v, 3) < \theta(v, 0, 2))$). Likewise, for the “ $1 \rightarrow 3$ ” transition to occur at v , the number of neighbors of v in state 2 or state 3 must be at least $\theta(v, 1, 3)$ (i.e., $(N(v, 2) + N(v, 3) \geq \theta(v, 1, 3))$). At any state $j \in \{0, 1, 2, 3\}$, if none of the conditions for transitions out of j hold, the node remains in state j .

The above general model is powerful as it allows the two contagions to interact. Many references have considered cooperating and competing contagions (e.g., [10, 12, 15]). For example, in the case of cooperating contagions, if a node has already contracted \mathbb{C}_1 , it may be easier for it to contract \mathbb{C}_2 . This can be modeled by choosing a low value for $\theta(v, 1, 3)$. However, the model is also complex since it requires the specification of five threshold values for each node. In this paper, we consider a simpler model which uses only two threshold values for each node.

A Simpler Form of Local Functions: In the general form discussed above, each node was associated with five threshold values, one corresponding to each of the five transitions shown in Fig. 1. In the simpler model, for each node v , we use two threshold values, denoted by $\theta(v, 1)$ and $\theta(v, 2)$. The parameter $\theta(v, 1)$ is used when v is in state 0 or state 2 (i.e., has not contracted contagion \mathbb{C}_1); it specifies the minimum number of neighbors of v whose state is either 1 or 3 so that v can contract contagion \mathbb{C}_1 . Similarly, $\theta(v, 2)$ is used when v is in state 0 or 1, and it specifies the minimum number of neighbors of v whose state is either 2 or 3 so that v can contract contagion \mathbb{C}_2 . Unlike the general model, the simpler model does not permit other interactions between the two contagions. However, the simpler model facilitates the development of analytical and experimental results.

Additional Definitions Concerning SyDSs: At any time τ , the **configuration** \mathcal{C} of a SyDS is the n -vector $(s_1^\tau, s_2^\tau, \dots, s_n^\tau)$, where $s_i^\tau \in \mathbb{B}$ is the state of node v_i at time τ ($1 \leq i \leq n$). Given a configuration \mathcal{C} , the state of a node v in \mathcal{C} is denoted by $\mathcal{C}(v)$. As mentioned earlier, in a SyDS, all nodes compute and

update their next state *synchronously*. Other update disciplines (e.g., sequential updates) have also been considered in the literature (e.g., [2,13]). Suppose a given SyDS transitions in one step from a configuration \mathcal{C}' to a configuration \mathcal{C} . Then we say that \mathcal{C} is the **successor** of \mathcal{C}' , and \mathcal{C}' is a **predecessor** of \mathcal{C} . Since the SyDSs considered in this paper are deterministic, each configuration has a *unique* successor. However, a configuration may have zero or more predecessors. A configuration \mathcal{C} which is its own successor is called a **fixed point**. Thus, when a SyDS reaches a fixed point, no further state changes occur at any node.

Example: The underlying network of a SyDS in which two contagions are propagating under the simpler model discussed above is shown in Fig. 2.

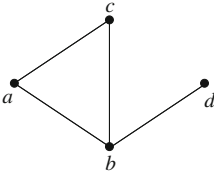


Fig. 2. The underlying network of a SyDS with two contagions. For each node v , both the threshold values are 1.

For each node v , the two threshold values $\theta(v, 1)$ and $\theta(v, 2)$ are both chosen as 1. Suppose the initial states of nodes a, b, c and d are 1, 2, 0 and 0 respectively; that is, the initial configuration of the system is $(1, 2, 0, 0)$. The local function f_a at a is computed as follows. Since a is in state 1, we need to check if it can contract contagion \mathbb{C}_2 . Since $\theta(a, 2) = 1$ and a has a neighbor (namely b) in state 2, a can indeed contract contagion \mathbb{C}_2 .

Therefore, the value of the local function f_a is 3; that is, the next state of a is 3. In a similar manner, it can be seen that the local functions f_b and f_c (at nodes b and c respectively) also evaluate to 3. For node d , whose current state is 0, there is one neighbor (namely, b) whose state is 2. Therefore, the local function f_d at d evaluates to 2. Thus, the configuration of the system at time 1 is $(3, 3, 3, 2)$. Since the system is progressive, the states of nodes a, b and c will continue to be 3 in subsequent time steps. However, the state of node d changes to 3 at time step 2 since d has a neighbor (namely, b) whose state at time step 1 is 3. Thus, the configuration of the system at the end of time step 2 is $(3, 3, 3, 3)$. In other words, the sequence of configurations at times 0, 1 and 2 of the system is:

$$(1, 2, 0, 0) \longrightarrow (3, 3, 3, 2) \longrightarrow (3, 3, 3, 3)$$

Once the system reaches the configuration $(3, 3, 3, 3)$, no further state changes can occur. Thus, the configuration $(3, 3, 3, 3)$ is a fixed point for the system.

In this example, the SyDS reached a fixed point. Using our assumption that the system is progressive, one can show that every such SyDS reaches a fixed point.

Proposition 1. *Every progressive SyDS under the two contagion model reaches a fixed point from every initial configuration.*

Proof: Consider any progressive SyDS on $\mathbb{B} = \{0, 1, 2, 3\}$. Let n denote the number of nodes in the underlying graph of the SyDS. In any transition from a configuration to a different configuration, at least one node changes state.

Because the system is progressive, each node may change state at most twice: once from 0 to 1 (or 0 to 2) and then from 1 to 3 (or 2 to 3). Thus, after at most $2n$ transitions where the states of one or more nodes change, there can be no further state changes. In other words, the system reaches a fixed point after at most $2n$ transitions. ■

Problem Formulation: The focus of this paper is on a method for containing the propagation of two simultaneous contagions by appropriately vaccinating a subset of nodes. Before defining the problem formally, we state the assumptions used in our formulation.

Following [11], we assume that only those nodes that are initially uninfected by either contagion (i.e., nodes whose initial state is 0) can be vaccinated for \mathbb{C}_1 and/or \mathbb{C}_2 . When a node is vaccinated for a certain contagion, the node cannot get infected by that contagion; as a consequence, such a node cannot propagate the corresponding contagion. For $i = 1, 2$, one can think of vaccinating a node v for a contagion \mathbb{C}_i as increasing the threshold $\theta(v, i)$ of the node v to $\text{degree}(v)+1$ so that the number of neighbors of v that are infected by \mathbb{C}_i will always be *less than* $\theta(v, i)$. If a node v is vaccinated for both \mathbb{C}_1 and \mathbb{C}_2 , then it plays no role in propagating either contagion. In such a situation, one can think of the effect of vaccination as removing node v and all the edges incident on v from the network.

The optimization problem studied in this paper is a generalization of a problem studied in [11] for a single contagion. This problem deals with choosing a small set of nodes to vaccinate so that the total number of resulting new infections when the system reaches a fixed point is a minimum. Given a set C of nodes to be vaccinated, a **vaccination scheme** specifies for each node $w \in C$, whether w is vaccinated against \mathbb{C}_1 , \mathbb{C}_2 or both. The total number of vaccinations used by a vaccination scheme for a set of nodes C is the sum $N_1 + N_2$, where N_i is the number of nodes vaccinated against \mathbb{C}_i , $i = 1, 2$. Note that if a node w is vaccinated against both \mathbb{C}_1 and \mathbb{C}_2 , then it is included in both N_1 and N_2 . Also, after a vaccination scheme is chosen and the contagions spread through a network, the number of new infections is measured as the total number of state transitions, because each state transition means a node acquires a new contagion. A formal statement of this optimization problem is as follows.

Vaccination Scheme to Minimize the Total Number of New Infections (VS-MTNNI)

Given: A social network represented by the SyDS $\mathbb{S} = (G, \mathbb{F})$ over $\mathbb{B} = \{0, 1, 2, 3\}$, with each local function $f_v \in \mathbb{F}$ at node v represented by two threshold values $\theta(v, 1)$ and $\theta(v, 2)$; the set I of **seed** nodes which are initially infected (i.e., the state of each node in I is from $\{1, 2, 3\}$); an upper bound β on the total number of vaccinations.

Requirement: A set $C \subseteq V - I$ of nodes to be vaccinated and a vaccination scheme for C so that (i) the total number of vaccinations is at most β and (ii) among all subsets of $V - I$ which can be vaccinated to satisfy (i), the set C and the chosen vaccination scheme lead to the smallest number of newly infected nodes.

It is straightforward to show that VS-MTNNI is computationally intractable. To do this, we state a problem and a result from [11].

Smallest Critical Set to Minimize the number of Newly Affected Nodes (SCS-MNA)

Given: A SyDS represented by a graph $G(V, E)$ through which a single contagion is propagating, a threshold value $\theta(v)$ for each node v , a set $I \subseteq V$ of initially infected nodes, a vaccination budget β and an upper bound Q on the number of new infections.

Requirement: A subset $C \subseteq V$ such that $|C| \leq \beta$ and after vaccinating the nodes in C , the number of new infections in G is at most Q .

The following result is from [11].

Theorem 1. *The SCS-MNA problem is NP-hard even when each threshold value is 2. Further, if the vaccination budget cannot be violated, the problem cannot be approximated¹ to within any factor $\rho \geq 1$, unless $P = NP$. ■*

It is easy to show that the result of Theorem 1 also holds for the VS-MTNNI problem.

Proposition 2. *The VS-MTNNI problem is NP-hard even when each threshold value is 2. Further, if the vaccination budget cannot be violated, the problem cannot be approximated to within any factor $\rho \geq 1$, unless $P = NP$.*

Proof: The SCS-MNA problem can be easily reduced to the VS-MTNNI problem as follows. Let an instance of SCS-MNA be given by a graph $G(V, E)$, a subset $I \subseteq V$ of initially infected nodes (by the only contagion), a vaccination budget β and an upper bound Q on the number of new infections. From the graph $G(V, E)$ of the SCS-MNA instance, we create a new graph $G'(V', E)$ by adding a new node v to V such that v has no edges incident on it. In the VS-MTNNI instance, the initial state of each node in I is chosen as 1 and the initial state of the new node v is chosen as 2. The two threshold values for each node in G' are chosen as 2. It is now easy to see that only \mathbb{C}_1 can spread in the SyDS represented by G' . Therefore any vaccination scheme for G' which vaccinates at most β that causes at most Q new infections is also a solution to the SCS-MNA instance, and vice versa. ■

Proposition 2 points out that in the worst-case, even obtaining an efficient approximation algorithm with a provable performance guarantee for the VS-MTNNI problem is computationally intractable. Therefore, we now focus on designing a heuristic algorithm that works well in practice. This heuristic relies on a known approximation algorithm for a generalized version of the Set Cover problem, called the Set Multicover problem [20]. In this problem,

¹ An algorithm for the SCS-MNA problem provides a factor ρ approximation if for every instance of the problem, the number of new infections is at most ρQ^* , where Q^* is the minimum number of new infections.

we are given a universal set $U = \{u_1, u_2, \dots, u_n\}$ of elements, a collection $C^* = \{C_1, C_2, \dots, C_m\}$ of subsets of U , an integer coverage requirement $r_i \geq 1$ for each $u_i \in U$, $1 \leq i \leq n$, a budget $\beta \leq m$. The goal is to find a subcollection $C' \subseteq C^*$ such that $|C'| \leq \beta$ and for each $u_i \in U$, the number of sets in C' that contain u_i is at least r_i , $1 \leq i \leq n$. When $r_i = 1$, $1 \leq i \leq n$, then we have the usual Set Cover problem [7]. An iterative greedy heuristic (which in each iteration picks a set which covers the largest number of elements whose coverage requirement has not yet been met) is known to provide a performance guarantee of $O(\log n)$ for the Set Multicover problem [20]. As discussed in Sect. 3 this heuristic is useful in developing our heuristic for the VS-MTNNI problem.

3 Experimental Results

In this section, we provide the networks tested; descriptions of the key elements of the analysis process—simulation and the contagion blocking heuristics (including the new MCICH); a summary of the overall analysis steps; and results of the contagion blocking numerical experiments. Throughout this section, we use the words “activated” and “infected” as synonyms, and also “block” and “vaccinate” as synonyms.

Networks: The three networks of Table 3 are evaluated. We use only the giant components from the networks.

Table 3. Networks used in experiments, and selected properties. All properties are for the giant component of each graph. These properties were computed using the `net.science` system [1].

Network	Num. nodes	Num. edges	Ave. degree	Ave. clust. coeff	Diameter
Astroph	17,903	196,972	22.0	0.633	14
FB-Politicians	5,908	41,706	14.1	0.385	14
Wiki	7,115	100,762	28.3	0.141	7

Simulation Process: A **simulation** consists of a set of iterations. Each **iteration** consists of software execution of contagion propagation from a **seed set** I , where seed nodes states are 1, 2, or 3. The total number of seed nodes is 20 in all iterations, and are chosen from the 20-core of each graph. (The 20-core of a graph G is the subgraph of G in which every node has a degree of at least 20 [6].) Each of the seed nodes has a probability of 1/3 of being set to each of states 1, 2, and 3. (All iterations were also done with 10 seed nodes, but results are not reported here.) An iteration starts at $t = 0$ with the seed nodes as the only activated nodes. From these nodes, contagion propagates in discrete times $t \in [1 .. t_{max}]$ as described for the SyDS in Sect. 2. All state transitions, x to y , are recorded for all $v \in V$. In this work, all iterations within one simulations use *uniform* thresholds for all nodes and all state transitions, so we abbreviate the thresholds below by setting $\theta = \theta(v, 1) = \theta(v, 2)$. In this work, we run 10

iterations per simulation, where the differences among the iterations is the composition of the seed node sets. Simulations are run with and without blocking nodes.

Blocking Heuristics: We present three methods (heuristics) for blocking a contagion. The first two are well studied, and serve as baselines for comparison. The third method is the covering heuristic MCICH that is a contribution of this work. For a simulation involving two distinct contagions, the corresponding method is applied for each contagion individually.

Random Heuristic. For a given budget β_i on the number of blocking nodes for contagion \mathbb{C}_i , select β_i nodes from among all nodes, uniformly at random.

High Degree Heuristic. For a given budget β_i on the number of blocking nodes for contagion \mathbb{C}_i , select the β_i nodes with the greatest degrees (break ties arbitrarily).

New Multi-Contagion Independent Covering Heuristic (MCICH). We devise a set cover heuristic to identify a subset of nodes that are activated at time t , to set as blocking nodes, such that no nodes will activate at time $t + 1$. If this is accomplished, then the contagion is halted at t , and our goal is achieved.

A key idea is that any node v_i that is activated at time $t + 1$ does so because it receives influence from nodes activated at time t , for otherwise, v_i would have activated at an earlier time. Thus, for a node v_i that gets activated at time $t + 1$, vaccinating or blocking nodes at time t will halt contagion propagation to v_i . This idea is used in the algorithm as follows. Consider the sets S_t and S_{t+1} of nodes that get infected or activated at times t and $t + 1$, respectively. We identify nodes from S_t , one at a time, iteratively, where the node v_k that is removed from S_t has the most edges in the graph G to nodes that are still infected in S_{t+1} . Each time a v_k is removed from S_t , the “covering requirement” for each neighbor $v_j \in S_{t+1}$ is reduced by 1, and when v_j 's requirement is 0, by removal of one or more nodes from S_t , that means v_j can no longer be infected for contagion \mathbb{C}_i .

The algorithm for the MCICH is presented in Algorithm 1. The algorithm computes the set C of blocking nodes for contagion \mathbb{C}_i for one iteration.

Summary of Analysis Process: The steps of the full analysis follow. Step 1: simulations are performed without consideration of blocking nodes, as described above. Step 2: using the simulation outputs, blocking nodes are determined using the blocking heuristics and specified blocking node budget β_i for contagion \mathbb{C}_i . Step 3: the simulations are repeated, with all conditions the same as in Step 1, except that now the blocking nodes are added (these blocking nodes remain in state 0). Note that the simulation and blocking methods, models and codes can handle—as they currently exist—non-uniform thresholds across nodes, different thresholds per contagion for each node, and heterogeneities in other parameters. We are reporting uniform threshold results owing to space limitations and because it is important to understand baseline behaviors.

Simulation and Blocking Results: Unless otherwise stated, all results are averages over all 10 iterations of a simulation.

Algorithm 1: Steps of the node blocking algorithm MCICH.

- 1 **Input:** Threshold $\theta = \theta(v, i)$ for contagion \mathbb{C}_i . A network $G(V, E)$. A set I of initially activated nodes (at time $t = 0$). Budget β_i on the number of blocking nodes for contagion \mathbb{C}_i . Maximum number t_{max} of time steps to run simulation.
 - 2 **Output:** The set C of blocking nodes such that $|C_i| \leq \beta_i$ and such that the number of (newly) activated nodes is small.
 - 3 **Steps:**
 - (A) Run simulation of contagion propagation.
 - (i) Compute the activated nodes at each time step from $t = 1$ through t_{max} .
 - (ii) The output is a set S_t of newly activated nodes at each time $t \in \{0, 1, 2, \dots, t_{max}\}$, where $S_0 = A$.
 - (B) Run the MCICH to obtain blocking node set C .
 - (i) **for** $t = 1$ to t_{max} :
 - (1) **if** $|S_t| \leq \beta_i$ **then** set $C = S_t$ and **return** C . Stop.
 - (2) Initialize the candidate set of blocking nodes T_t for this t to $T_t = \emptyset$.
 - (3) Set $Q_{t+1} = S_{t+1}$; Q_{t+1} 's elements will be removed iteratively.
 - (4) **for each** $v_k \in S_{t+1}$, compute the number ρ_k of neighbors that must be *un*-activated in order to prevent v_k from being activated. Here, $\rho_k = n_k - \theta_k + 1$, where n_k is the number of neighbors of v_k in G that are activated at any t^* , $0 \leq t^* < t$.
 - (5) **while** Q_{t+1} not empty **and** $|T_t| < \beta_i$ **do**:
 - (a) **for each** $v_j \in S_t$, let H_j be the subset of nodes in S_{t+1} for which v_j is a neighbor in G .
 - (b) Select the node v_k such that $\max_k |H_k \cap Q_{t+1}|$. Break ties arbitrarily.
 - (c) Add v_k to T_t , the candidate set of blocking nodes.
 - (d) For each node v_j in H_k , reduce ρ_j by 1. **if** $\rho_j = 0$ **then** remove v_j from all H_k **and** remove v_j from Q_{t+1} .
 - (6) **if** Q_{t+1} is empty **then** set $C = T_t$ and **return** C . Stop.
 - (ii) No blocking set was found to completely stymie the contagion. Iterate through all Q_{t+1} for all $t \in [1 .. t_{max} - 1]$ and set $C = T_t$ for the smallest $|Q_{t+1}|$; if ties, choose the one at the earliest t . **Return** C . Stop.
-

Basic Simulation Data and Temporal Blocking Effects. Figure 3 provides three types of results for the FB-Politicians network. The first two plots show temporal data on the spread or propagation of both contagions \mathbb{C}_1 and \mathbb{C}_2 simultaneously *without* blocking. The third plot shows temporal effects of blocking nodes on the propagation of both contagions. Figure 3a shows the number of newly activated nodes at each time step. The curves rise as uniform threshold decreases from 4, to 3, to 2, since contagion propagates more readily for lesser thresholds. Figure 3b shows the corresponding plots of *total* or cumulative number of nodes activated for both contagions as a function of time. Roughly 40% to 70% of FB-Politicians nodes are activated by $t_{max} = 24$, depending on θ . Figure 3c uses the $\theta = 3$ data from Fig. 3b as a baseline, and shows three additional curves, one for each of the three blocking methods discussed above. These data show that for a blocking budget $\beta_i = 0.02$ fraction of nodes, the MCICH performs best (i.e., the curve is the lowest). For blocking contagions “lesser” (or “lower”) is better. However, this budget is not sufficient to completely halt the contagion.

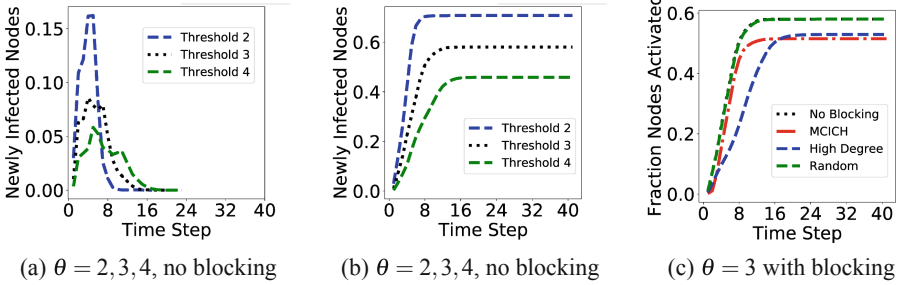


Fig. 3. Simulation results for the FB-Politicians network, where results are averages over 10 iterations. (a) shows time histories of the average number of *newly* activated nodes at each time step for contagions \mathbb{C}_1 and \mathbb{C}_2 combined, for three thresholds. (b) shows time histories of the average number of *cumulative* activated nodes at each time step for contagion \mathbb{C}_1 and \mathbb{C}_2 combined, for the same thresholds. (c) provides data for $\theta = 3$, for no blocking, and for each of the three blocking methods, where the blocking node budget $\beta_i = 0.02$ fraction of nodes. No method completely blocks the contagion (a greater budget is required), but MCICH performs best over the entire time history.

Efficacy and Comparisons of All Blocking Methods Across All Networks.

Figure 4 depicts the efficacy of all of the blocking methods for the three networks, for threshold values $\theta = 2, 3$, and 4. Data for one network are in a row, and data for one threshold are in one column. Each plot presents the cumulative fraction of activated nodes, as a function of the blocking budget in terms of fraction of network nodes. Note that the y-axis is the total number of activations, so that, for example, if a node has contracted \mathbb{C}_1 and \mathbb{C}_2 , then that counts as two activations. The cumulative fraction of activated nodes corresponds to the points at t_{max} in curves such as those presented in Fig. 3c, for the respective blocking methods, thresholds, and networks. There is a “no blocking” curve, and three curves for each of the random blocking nodes heuristic, high degree blocking nodes heuristic, and MCICH in each plot. Since lower curves represent more effective blocking, it is clear that MCICH performs far better, in the great majority of cases, than do the random and high-degree blocking heuristics. The blocking budget β is currently allocated between the two contagions using proportion of nodes infected by contagions \mathbb{C}_1 and \mathbb{C}_2 when there is no blocking. For example, suppose n_1 and n_2 denote the number of newly infected nodes by \mathbb{C}_1 and \mathbb{C}_2 respectively, we use $n_1/(n_1 + n_2)$ fraction of the budget for blocking \mathbb{C}_1 and the remaining budget for \mathbb{C}_2 . If the algorithm needs less than the allocated budget for blocking \mathbb{C}_1 , the remaining allocation is used to increase the budget for \mathbb{C}_2 .

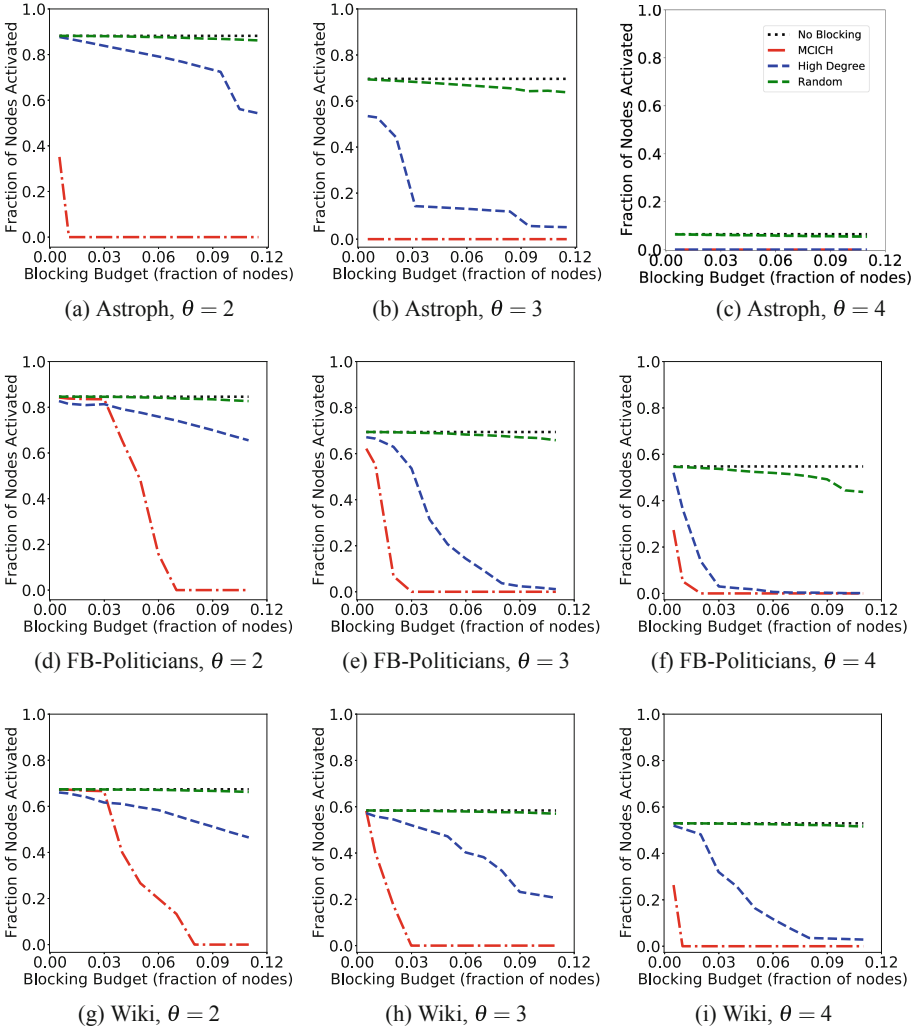


Fig. 4. Simulation and blocking results of applying all three blocking methods to block two-contagion spreading in three networks, using different threshold values for contagion propagation for each network. Results for each network are in one row. From the top to bottom rows, the networks are: Astroph, FB-Politicians, and Wiki. Each column contains data one threshold: left to right, $\theta = 2, 3,$ and 4 . Each plot displays the fraction of nodes contracting either contagion in diffusion simulations, as a function of the fraction of nodes used as blocking nodes, employed to stop the contagions. In each plot, there are four curves. Contagion spreading without blocking (black dots) is the reference curve, and is a horizontal line. Results from the random selection of blocking nodes is the green curve. Results from selecting the highest degree nodes as blocking nodes is the blue dashed curve. Results from MCICH method is the red dash-dot curve. The lower the curve, the better the performance in blocking contagion. The MCICH method does significantly better in all cases.

4 Future Research Directions

There are several directions for future work. For example, it is of interest to evaluate the MCICH heuristic under several other scenarios; examples include graphs with non-uniform threshold values for nodes, different ways of selecting seed sets and skewed distributions of seed nodes between the two contagions. It is also of interest to investigate the sensitivity of our heuristic with respect to the choice of seed sets. In our model, the two contagions are independent. It is of interest to investigate models where the contagions interact; that is, a node that is infected one contagion may make it easier or harder for the node to be infected by the other contagion.

Acknowledgments. We thank the reviewers for their comments. This work is partially supported by NSF Grants ACI-1443054 (DIBBS), IIS-1633028 (BIG DATA), CMMI-1745207 (EAGER), OAC-1916805 (CINES), CCF-1918656 (Expeditions), CRISP 2.0-1832587 and IIS-1908530.

References

1. Ahmed, N.K., Alo, R.A., Amelink, C.T., et al.: net.science: a cyberinfrastructure for sustained innovation in network science and engineering. In: Gateway (2020)
2. Barrett, C., Hunt III, H.B., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E., Thakur, M.: Predecessor existence problems for finite discrete dynamical systems. *Theor. Comput. Sci.* **386**(1), 3–37 (2007)
3. Beutel, A., Prakash, B.A., Rosenfeld, R., Faloutsos, C.: Interacting viruses in networks: can both survive? In: KDD, pp. 426–434 (2012)
4. Centola, D., Macy, M.: Complex contagions and the weakness of long ties. *Am. J. Sociol.* **113**(3), 702–734 (2007)
5. Chakrabarti, D., Wang, Y., Wang, C., Leskovec, J., Faloutsos, C.: Epidemic thresholds in real networks. *TISSEC* **10**, 1–33 (2008)
6. Easley, D., Kleinberg, J.: *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press (2010)
7. Garey, M.R., Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman & Co., San Francisco (1979)
8. González-Bailón, S., Borge-Holthoefer, J., Rivero, A., Moreno, Y.: The dynamics of protest recruitment through an online network. *Sci. Rep.* **1**, 7 (2011)
9. Granovetter, M.: Threshold models of collective behavior. *Am. J. Sociol.* 1420–1443 (1978)
10. Karrer, B., Newman, M.E.J.: Competing epidemics on complex networks. *Phys. Rev. E* **84**(3) (2011). <https://doi.org/10.1103/PhysRevE.84.036106>
11. Kuhlman, C.J., Kumar, V.A., Marathe, M.V., Ravi, S., Rosenkrantz, D.J.: Inhibiting diffusion of complex contagions in social networks: theoretical and experimental results. *Data Min. Knowl. Disc.* **29**(2), 423–465 (2015)
12. Kumar, P., Verma, P., Singh, A., Cherifi, H.: Choosing optimal seed nodes in competitive contagion. *Front. Big Data* **2**, 1–6 (2019)
13. Mortveit, H., Reidys, C.: *An Introduction to Sequential Dynamical Systems*. Springer, New York (2007)

14. Myers, S.A., Leskovec, J.: Clash of the contagions: cooperation and competition in information diffusion. In: ICDM, pp. 539–548 (2012)
15. Newman, M.E.J., Ferrario, C.R.: Interacting epidemics and coinfection on contact networks. *PLoS ONE* **8**, 1–8 (2013)
16. Romero, D., Meeder, B., Kleinberg, J.: Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In: WWW, pp. 695–704 (2011)
17. Schelling, T.C.: *Micromotives and Macrobehavior*. Norton, New York (1978)
18. Stanoev, A., Trpevski, D., Kocarev, L.: Modeling the spread of multiple concurrent contagions on networks. *PLoS ONE* **9**(6), e95669 (2014)
19. Ugander, J., Backstrom, L., Marlow, C., Kleinberg, J.: Structural diversity in social contagion. *Proc. Natl. Acad. Sci.* **109**(16), 5962–5966 (2012)
20. Vazirani, V.V.: *Approximation Algorithms*. Springer, Heidelberg (2001)
21. Watts, D.J.: A simple model of global cascades on random networks. *Proc. Natl. Acad. Sci.* **99**, 5766–5771 (2002)



Stimulation Index of Cascading Transmission in Information Diffusion over Social Networks

Kazufumi Inafuku¹(✉), Takayasu Fushimi², and Tetsuji Satoh¹

¹ University of Tsukuba, Tsukuba, Ibaraki 305-8550, Japan
{inafuku,satoh}@ce.slis.tsukuba.ac.jp

² Tokyo University of Technology, Hachioji, Tokyo 192-0982, Japan
fushimity@stf.teu.ac.jp

Abstract. Analyzing and modeling of information diffusion on social networks is essential because social networking sites (SNSs) have become crucial information infrastructures. In particular, “Influence Maximization,” the extraction of information source nodes that deliver information to as many users as possible on a network, has been widely researched. However, actual information diffusion is caused not only propagation according to the network structure, but also a local rise in “trending” topics. We therefore focused on the edges that cause a chain of information transmission, regardless of the number of people who received the information. Based on the information cascade, where information is propagated in chains between nodes on a network, we propose the Stimulation Index to quantify how much edges affect the subsequent transmission of information. We also evaluate the proposed index using an artificial network and verify that it is effective.

Keywords: Information diffusion · Information cascade · Social networks

1 Introduction

Social networking services (SNSs) such as Twitter and Facebook have become crucial information infrastructures, transmitting various information, such as news and rumors, from person to person. Thus, it is important research question to understand the process of information diffusion and how many people it reaches.

In viral marketing (the promotion of products by word of mouth on social networks), there is a need to reach as many users as possible within a limited budget. This problem can be considered an estimation of the essential users who can convey information to the most users and is called the “Influence Maximization [7].” In addition, in times of disaster, lies and rumors spread, and many users receive false information and transmit it. There are also studies on how to disseminate corrections and block the spread of false information efficiently [6, 8].

These Influence Maximization studies focus on the number of users information reaches. However, in a social network in which the dissolution and merging of communities are commonplace and change daily, it is important to consider not only the range and scale of information diffusion, but also the quality of information diffusion; that is, “how much the topic is talked about” within a local community. For example, Twitter trends pick up not only topics interesting to many users, but also topics that are exciting among a small group of users. In addition to propagating information through the network structure, users receive information from trends and, as a result, information reaches a large number of users. The edge, which causes a lot of information transfer, is considered to play an essential role in information diffusion, as is the user (node) who delivers information to many users.

In this study, our goal is to detect important edges in information diffusion on social networks. In information diffusion, a node that receives information transmits it to out-neighbor nodes. We focus on the transmission to multiple nodes, i.e., which edge influences topic excitement. When the amount of received information exceeds a certain threshold, information is transmitted to out-neighbor nodes. Based on the information cascade phenomenon, in which information is propagated as a chain, we propose the “Stimulation Index” as an edge’s importance index of information diffusion. The Stimulation Index quantifies how much an edge affects subsequent information transmission.

The contributions of this study are as follows. First, we propose the “Stimulation Index” to quantify information diffusion’s importance in social networks. Second, we test our approach by attempting to detect high Stimulation Index edges using an artificial network and confirm that these edges have an important role in information diffusion.

2 Related Work

2.1 Information Diffusion Model

When information is transmitted from one person to another, it is not only exchanged between two people, but is also spread more widely as the receiver transmits it to new people. This information transfer chain phenomenon is called the “Information Cascade,” [1] and there has been much research on modeling this phenomenon on complex networks. In particular, the “Independent Cascade model (IC model)” [4, 7] and “Linear Threshold model (LT model)” [15, 16] are widely used.

In comparing these two models, the IC model is sender centric. The diffusion probability must be assigned to each edge in advance. We then select an information source node and the node transmits information to the neighboring nodes. The success or failure of information transmission depends independently on each edge’s diffusion probability. If the information transmission is successful, the receiver node becomes the sender node and receiving information should be sent to the neighboring nodes. The IC model emulates information diffusion by repeating this process.

On the other hand, the LT model simulates information diffusion around the receiver nodes. First, we assign a threshold (0 to 1) for activation to each node. We then choose a source node, which transmits the information to all neighboring nodes. Neighboring nodes receive $\frac{1}{in-degrees}$ as weights. When the sum of the received weights exceeds the threshold value, the receiver node is activated and transmits the information to neighboring nodes. These models have been used, for example, in “Influence Maximization” [7,9,11] to find nodes that maximize the expected number of nodes that receive information. Also, since real-world networks are dynamic, a dynamic extension of Influence Maximization [12,13] is also being studied.

2.2 Analysis and Estimation of Information Diffusion

Gomez [5] regarded the information diffusion path as a dynamic network and estimated it. Chen [3] made predictions about whether an information cascade will continue to grow photo sharing on Facebook. In addition, Kawamoto [14], in their study on early detection of socially influential information cascades on Twitter, focused on the content of information that spread and clarified the effects of text features on information diffusion. Yoshikawa [17] proposed an extension of the IC and LT models to estimate the expected influence curve on social networks. They used expectation-maximization (EM) algorithms to determine the sequence of information diffusion and estimated the expected influence curve by simulation using the learned model parameters.

3 Stimulation Index of Cascading Transmission

Social networks, including Twitter and mailing list networks, are known to create an “information cascade” in which information propagates in chains among users. This information cascade can be treated as a dynamic network by regarding users as nodes and information transmissions (such as posting to Twitter and sending emails) as edges. Information spreads over a wide area through network links when a user is interested in what is received and tries to spread it to other users; that is, one information transmission stimulates and induces subsequent information transmissions continuously. In this study, we attempt to quantify the importance of edges by the amount of information transmission, i.e., the appearance of subsequent edges. This section describes the basic idea and calculation method of the proposed index.

3.1 Basic Idea of Proposed Method

Consider a directed graph $G = (V, E)$, where V and E denote a set of nodes and edges, respectively. We represent an edge that appears at time t as e_t .

Figure 1(a) shows the simplest example of information cascades over a network with 4 nodes and 3 edges. At time $t = 1$, edge e_1 appeared from node v_a to v_b . Later, e_2 appeared at time $t = 2$. In this case, we consider that v_b was

stimulated by e_1 and e_2 was induced by e_1 . In the same way, we consider that e_2 inspired the occurrence of e_3 . Also, since e_3 can be regarded as the result of an information cascade from e_1 , we assume that e_1 also stimulated e_3 . Thus, the proposed method scores $s(e_t)$ for 3 edges as follows: $s(e_1) = 2$, $s(e_2) = 1$, and $s(e_3) = 0$.

Figure 1(b) represents an example of a new edge produced by stimulation of multiple edges, where e_3 occurs after e_1 and e_2 . Although, similarly to the above-mentioned case, e_1 inspired the appearance of e_3 and e_2 also inspired that of e_3 , their scores are divided by two since the occurrence of e_3 is the contribution of the two edges e_1 and e_2 . As a result, the proposed method scores $s(e_t)$ for 3 edges as follows: $s(e_1) = 0.5$, $s(e_2) = 0.5$, and $s(e_3) = 0$.

Figure 1(c) depicts an example where node v_b has two in-edges e_1 and e_3 , and two out-edges e_2 and e_4 . In this case, node v_b received some information and inspired node v_a at time $t = 1$, v_b then sent the information to its followers, and node v_c received it at $t = 2$. Subsequently, at $t = 3$, v_b received other information from node v_d and sent it to its followers and v_e then received it. In this case, we assume that the information transmission e_2 was inspired by e_1 and e_4 was inspired only by e_3 . By doing this, our proposed measure demonstrates that the effect diminishes over time and avoids higher scores for older edges. As a result, the proposed method scores $s(e_t)$ for 4 edges are as follows: $s(e_1) = 1$, $s(e_2) = 0$, $s(e_3) = 1$, and $s(e_4) = 0$.

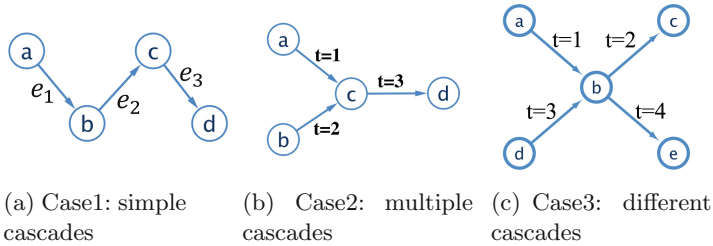


Fig. 1. A toy example of information cascades.

3.2 Calculation Method

The Stimulation Index $s(e_t)$ is defined as the sum of the directly stimulated score $sd(e_t)$ and indirectly stimulated score $si(e_t)$. By constructing an Edge-Relation (ER) graph EG that represents the relationship between the edges and dividing $s(e_t)$ into $sd(e_t)$ and $si(e_t)$, the score can be obtained efficiently.

Figure 2(a) shows the information diffusion network $G(V, E)$. As mentioned above, an edge may occur by inspiration from older edges, e.g., e_2 and e_3 were inspired by e_1 . By taking this assumption into consideration, we construct an ER graph as shown in Fig. 2(b). In Fig. 2, the stimulation of e_1 may affect many subsequent edges, but it directly affects only e_2 and e_3 ; thus, in the ER graph,

e_1 links to e_2 and e_3 . In this way, we construct ER graph $EG = (E, R)$, where nodes are edges E in the original graph $G = (V, E)$ and edges are pairs of edges that directly stimulate the relationship, $R \subset E \times E$. As can be seen in Fig. 2(b), the ER graph is a directed acyclic graph (DAG). Exploiting this ER graph, we calculate $s(e_t)$ according to Algorithm 1.

For node e_t in EG , we define sets of in-neighbor nodes and out-neighbor nodes as $IV(e_t)$ and $OV(e_t)$, respectively. First, the directly stimulated score of e_t is calculated as $sd(e_t) = \sum_{e \in OV(e_t)} \frac{1}{|IV(e)|}$. In Algorithm 1, the processes from line 3 to line 8 indicate the calculation of $sd(e_t)$.

Second, the indirectly stimulated score of e_t is calculated as $si(e_t) = \sum_{e \in OV(e_t)} \frac{sd(e) + si(e)}{|IV(e)|}$. Although, in order to compute $si(e_t)$, we need the indirectly stimulated scores $si(e_u)$, $u > t$ of edges later than t , we can calculate efficiently by accessing the nodes in order from the bottom due to its DAG struc-

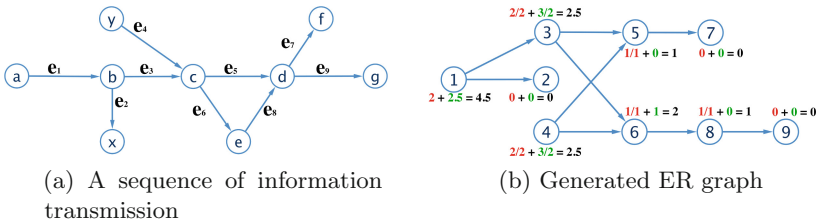


Fig. 2. A toy example: calculation of Stimulation Index for each edge. In this example, values colored as red are the directly stimulated scores, green are the indirectly stimulated scores, and black are stimulation indices.

Algorithm 1: Calculation of Stimulation Index

Data: Edge Relationship Graph $EG(E, R)$

Result: score dictionary

```

1 //calculate  $sd(e_t)$ 
2 set dictionary that keys and values are nodes and scores
3 for each node  $e_t$  of  $E$  do
4   get out-neighbors of  $e_t$ 
5   set  $sd(e_t) = 0$ 
6   for each node  $e_u$  of  $e_t$ 's out-neighbors do
7      $sd(e_t) += 1/\text{in-degrees of } e_u$ 
8    $dictionary[e_t] = sd(e_t)$ 
9 //calculate  $si(e_t)$ 
10 for each node  $e_t$  of  $E$  Order by  $t$  DESC do
11   get in-neighbors of  $e_t$ 
12   for each node  $e_u$  of  $e_t$ 's in-neighbors do
13      $dictionary[e_u] += dictionary[e_t] / \text{the number of } e_t\text{'s in-neighbors}$ 
14 return dictionary

```

ture, like the Brandes algorithm for betweenness centrality [2]. In Algorithm 1, the processes from line 10 to line 14 indicate the calculation of $si(e_t)$.

4 Experimental Settings

In order to evaluate the proposed method, we generated an artificial network that imitates a follow network and simulated an information diffusion process over the network. First, we generated an undirected network with a community structure using the LFR (Lancichinetti-Fortunato-Radicchi) benchmark graph. We next added a direction for each edge based on random walks. We then simulated information diffusion using an extended LT model to allow reactivation in a Susceptible-Infected-Susceptible manner.

4.1 Generation of Follow Network

To generate an artificial network that imitates directed follow networks equipped with the scale-free degree distribution and community structure, we employed the LFR benchmark graph [10], in which the degree and community size follow a power-law distribution.

Specifically, we generated an undirected network of 500 nodes, 1,138 edges, and 9 communities. In the parameter setting of the LFR model, the number of nodes is 500, the exponent of the degree distribution is 3, the exponent of community size distribution is 2, the average degree is 5, the minimum number of nodes in each community is 50, and the ratio of intra-community edges is 0.95.

Next, in order to add the direction for each edge, we conducted repeated random walks. Concretely, we selected the initial node u and randomly moved to a neighbor node v , repeating the move h times. Then, we set a directed edge $e = (u, v)$ according to the movement from u to v . When the same two-node combination appeared more than once, we adopted the direction with the higher number of occurrences. When the number of appearances was the same, we regarded it as a bidirectional edge. If an edge does not pass by a random walk, its direction is determined at random. We randomly selected five nodes from each community as initial nodes and set the number of movements as $h = 200$. As a result, we constructed a directed network of 500 nodes, 1,375 edges, and 9 communities.

4.2 Simulation of Information Diffusion

To produce an information diffusion sequence, we used the SIS (Susceptible-Infectious-Susceptible)-LT model, which is an extension of the LT model. In the usual LT model, the nodes have three states, Susceptible, Infected, and Recovered, based on the SIR model. In other words, once a node becomes active (infected) and sends information to its followers, it enters into the recovered state and never changes its state after that. In the elementary sense, however, users in an SNS send a variety of information to the same user repeatedly, unlike

in the case of infectious diseases, so we employed the SIS model for node state transitions. In the SIS model, an infected (active) node returns to a susceptible (inactive) state again after sending information. In our experiments, for each node, we updated the threshold of the LT model as a higher value than before reactivation at each time of reactivation; i.e., the more times a node enters the susceptible state, the higher the threshold for the next activation, which corresponds to “boredom with the topic.” Specifically, we represent the initial threshold and reactivation coefficients of node v as θ_v and α . Then, we set the value of the threshold in the n -th reactivation as $(n - 1)\theta_v + \alpha^n\theta_v$.

In our experiments, we simulated a biased information diffusion in which information transmission only occurs in a limited range of the network, such as an intra-community; that is, for certain information, a certain range of nodes actively transmits the information and other node groups do not transmit much or do not transmit at all. To generate biased information diffusion, the threshold θ_v is multiplied by the bias coefficient β_v . Let V_β be the set of nodes that actively spread information and set the bias coefficient β_v for $v \in V_\beta$ as follows:

$$\beta_v = \begin{cases} \frac{|V_\beta|}{|V|} & v \in V_\beta \\ \frac{|V|+|V_\beta|}{|V|} & v \notin V_\beta. \end{cases}$$

The value of coefficient β_v depends on the number of nodes in V_β . When $V_\beta = V$ holds, the values of β_v for all nodes $v \in V$ are 1.0, which is the same as the unbiased information diffusion in which all nodes evenly have the possibility to be active. In this way, we can produce biased artificial information diffusion sequences with almost the same number of information transmissions as unbiased ones.

We simulated information diffusion with the setting that each of all nodes $v \in V$ was treated as an information-source node. When the state of all nodes turns to inactive (susceptible), the simulation of information diffusion ends and we count the number of activations $as(v; G)$ and the number of activated nodes $ar(v; G)$. We conducted M times simulations, so we represent the values obtained by the m -th simulation as $as(v; G)^{(m)}$ and $ar(v; G)^{(m)}$, respectively. Then, we calculated the average values over all simulations as $\sigma_{as}(v; G) = 1/M \sum_{m=1}^M as(v; G)^{(m)}$ and $\sigma_{ar}(v; G) = 1/M \sum_{m=1}^M ar(v; G)^{(m)}$. Similarly, we also calculated the average values over all nodes as $\sigma_{as}(G) = 1/|V| \sum_{v \in V} \sigma_{as}(v; G)$ and $\sigma_{ar}(G) = 1/|V| \sum_{v \in V} \sigma_{ar}(v; G)$.

In Fig. 3, we show the visualization result of a generated network based on the LFR method, where the red-colored node is a node of V_β and $|V_\beta| = 51$. Using the network, we conducted $M = 10$ times simulations and, in each simulation, each of the $|V| = 500$ nodes is treated as an information source, thus the total number of information diffusion sequences is 5,000. We also set the reactivation coefficients to $\alpha = 1.1$.

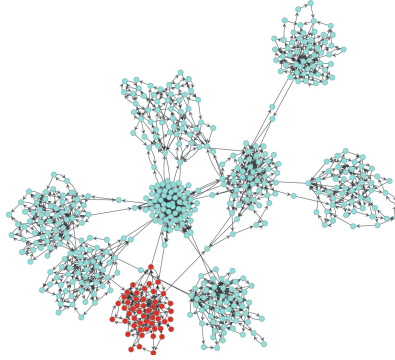


Fig. 3. Visualization result of generated network based on the LFR method.

5 Experimental Evaluations

5.1 Does Removing the Edge with a High Stimulation Index Inhibit Information Diffusion?

We evaluated whether the Stimulation Index can detect important edges in information diffusion using the artificial sequence generated in Sect. 4.

In our evaluation method, we remove any edge from the follow network G and extract subgraph G' . For this subgraph, we run an information diffusion simulation with the same settings. Now, the number of activations $\sigma_{as}(G')$ and the number of activated nodes $\sigma_{ar}(G')$ are lower than before the removal because we removed edges. The more these $\sigma_{as}(G')$ and $\sigma_{ar}(G')$ decrease, the more their edge deletions inhibit information diffusion; i.e., we consider them to be important edges in information diffusion. We confirmed that $\sigma_{as}(G')$ and $\sigma_{ar}(G')$ decrease more as the edge with a high Stimulation Index is removed, showing that the proposed index is effective.

However, the Stimulation Index is a measure of dynamic edges, while the following network is a static structure. Thus, edges connecting the same nodes may appear at different times t_x and t_y , such as $e_x = (u, v)$ and $e_y = (u, v)$. Then, we calculate the Stimulation Index for each static edge. We represent the dynamic graph for the m th simulation from any node v on G as $G_{v,m} = (V_{v,m}, E_{v,m})$ and the dynamic edge from node u to v that appeared at time t as $e_t = (u, v, t)$. We then define the Stimulation Index $si((u, v))$ of a static edge $(u, v) \in E$ as follows:

$$A_{v,m} = \{e_t \in E_{v,m} | e_t = (u, v, t)\}$$

$$si((u, v)) = \frac{\sum_{v \in V} \sum_{m=1}^M \sum_{e_t \in A_{v,m}} s(e_t)}{|V| \times M}.$$

We ranked the Stimulation Index of static edges using the information diffusion series simulated in Sect. 4. As a comparison method, we also calculated

the ranking of edge betweenness centrality and edge-degree centrality from the follow network G . We chose these indices because they are all representative centrality indicators and because we consider them to have an essential role in terms of information diffusion.

Figure 4 shows the number of activations $\sigma_{as}(G')$ and activated nodes $\sigma_{ar}(G')$ when removing the edge in each indices' ranking order. The horizontal axis corresponds to the edge deletion rate i , the vertical axes to metrics ($\sigma_{as}(G')$ and $\sigma_{ar}(G')$), and the plot-line to three edge indices. For example, the edge deletion rate $i = 0.1$ shows each metric value when removing the top 10% of edges in rankings. Naturally, as the deletion rate increases, information diffusion is inhibited, so the number of activations $\sigma_{as}(G')$ and activated nodes $\sigma_{ar}(G')$ decreases. Note that the edge deletion ratio of $i = 0.0$ means that no edges are removed, so we used the same simulation results for three rankings. Therefore, each metric has the same value. In Sect. 4, we simulated two types of thresholds θ_v for information diffusion in the SIS-LT model, one with a uniform distribution and the other with a biased distribution.

Figure 4(c) and Fig. 4(a) show the results of uniform distribution simulations. Although the Stimulation Index was smaller than the edge-degree centrality, it was not so different from the edge betweenness centrality. This result means that the edge's importance of information diffusion, such that information reaches the overall network equally, is not different from a static network case. Next, we focused on the results of biased simulations (Fig. 4(b) and Fig. 4(d)). Compared to the edge-degree centrality, the Stimulation Index shows that the number of activations $\sigma_{as}(G')$ and activated nodes $\sigma_{ar}(G')$ decreases at the step where the edge deletion rate i is low. Each metric ($\sigma_{as}(G')$ and $\sigma_{ar}(G')$) is also lower than the edge betweenness centrality, although the difference is somewhat smaller. The difference is especially noticeable in the case of $i \leq 0.05$, which shows that the important edges are ranked higher in the biased information diffusion. In other words, the proposed method works as expected.

5.2 Is There a Correlation Between the Stimulation Index and the Number of Activations and Activated Nodes?

In Sect. 5.1, we confirmed that removing edges with a high Stimulation Index decreases the number of activations $\sigma_{as}(G')$ and activated nodes $\sigma_{ar}(G')$. We reconsidered this result with a focus on nodes. Removing an edge (u, v) makes it difficult to activate the target node v because it cannot receive information from u . If v is a node with a high number of activations $\sigma_{as}(v; G')$ and activated nodes $\sigma_{ar}(v; G')$, it is evident that removing v will decrease them. Furthermore, if there is a high correlation between the Stimulation Index $s(u, v)$ and these metrics ($\sigma_{as}(v; G')$ and $\sigma_{ar}(v; G')$), the proposed index's effectiveness diminishes. Thus, this section confirms the correlation between $s(u, v)$ and these metrics ($\sigma_{as}(v; G')$, $\sigma_{ar}(v; G')$), and reinforces the experimental results' validity in Sect. 5.1.

First, Table 1 shows the correlation coefficients between Stimulation Index $s((u, v))$ and the number of activations $\sigma_{as}(v; G')$, and Stimulation Index

$s((u, v))$ and the number of activated nodes $\sigma_{ar}(v; G')$. There is no correlation in any of the experimental settings and metrics. Figure 5 shows the relationship between the Stimulation Index ranking of each edge and the number of activated nodes of the target node. Due to space limitations, we excluded the number of activations' results. The horizontal and vertical axis shows the Stimulation Index ranking and the number of activated nodes, respectively. Each plot point represents an edge or target node. We find some edges with a high ranking and a small number of activated nodes in both cases. Accordingly, the experimental results in Sect. 5.1 are valid.

Table 1. Pearson’s correlation coefficients for Stimulation Index and the number of activations, and Stimulation Index and the number of activated nodes

Metrics	Uniform simulation	Biased simulation
Number of activations	0.134	0.132
Number of activated nodes	0.135	0.122

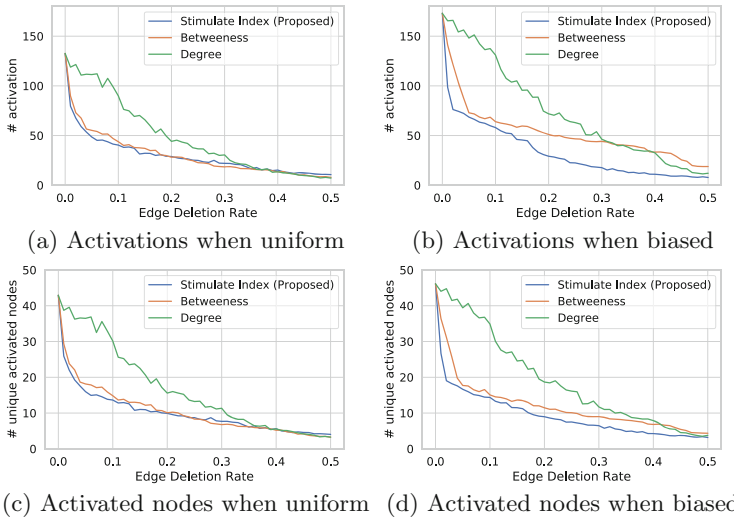


Fig. 4. The number of activations $\sigma_{as}(G')$ and activated nodes $\sigma_{ar}(G')$ when edges are removed.

6 Discussion

Addressing the edges with a high Stimulation Index and a low number of expected activations revealed in Sect. 5, Fig. 6 shows the network’s visualization results, reflecting the number of activations in the color of the edges. The

higher the number of activations, the darker the red. To focus on the edges with a high Stimulation Index, we extracted the top 30% edges in the ranking and colored the rest of the edges gray.

First, edges that connect communities have a high number of activations in both uniform and biased simulations. Figure 6(b) shows that edges in the community with a low threshold of information diffusion also have a high number of activations. On the other hand, many edges with a low number of activations often connect nodes in the same community to each other. These edges contribute to the transmission of information in the community. Moreover, these black edges can only be extracted by the Stimulation Index. In summary, the proposed index can extract the edges that induce information transmission in the community, in addition to the edges or nodes that deliver information to a larger number of users, which was the conventional method's focus.

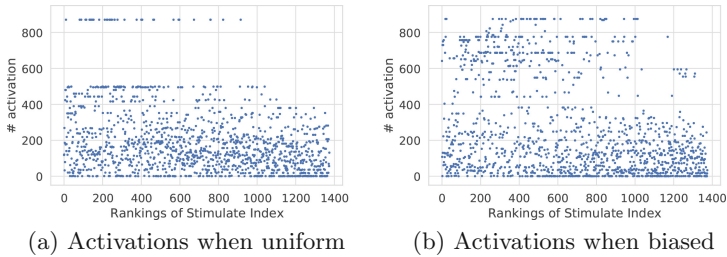


Fig. 5. Scatter plots of Stimulation Index and the number of activations.

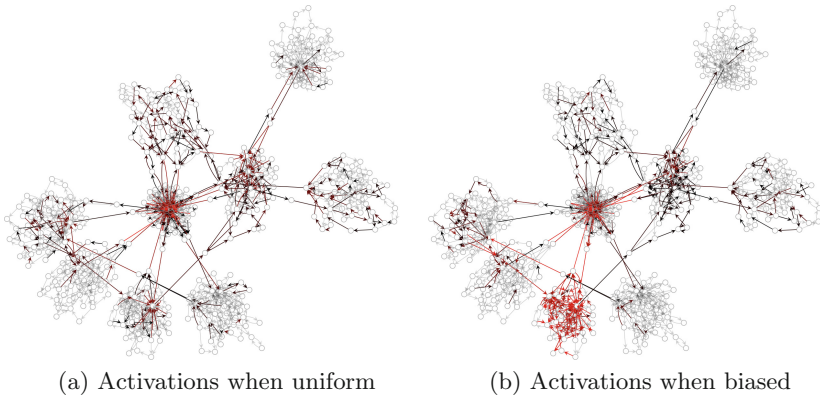


Fig. 6. The network's visualization results reflecting the number of activations in the color of the edges. The higher the number of activations, the darker the red.

7 Conclusion

In this study, we proposed a Stimulation Index, measuring an edge's spillover effect in information diffusion on networks. This Stimulation Index quantifies the amount of subsequent information transmission caused by an information transmission. To verify the proposed method's effectiveness, we simulated information diffusion using an artificial follow network. We demonstrated that removing edges with a high Stimulation Index inhibits information diffusion conspicuously.

Evaluation methods still require improvement. In this study, we adopted the number of activation and activated nodes to confirm the proposed index's effectiveness. However, it can be said that this is an indirect evaluation of the effects of the Stimulation Index. In order to more accurately evaluate the importance of information diffusion, we would like to examine other metrics.

References

1. Bikhchandani, S., Hirshleifer, D., Welch, I.: A theory of fads, fashion, custom, and cultural change as informational cascades. *J. Polit. Econ.* **100**(5), 992–1026 (1992)
2. Brandes, U.: A faster algorithm for betweenness centrality. *J. Math. Sociol.* **25**, 163–177 (2001)
3. Cheng, J., Adamic, L., Dow, P.A., Kleinberg, J.M., Leskovec, J.: Can cascades be predicted? In: *Proceedings of the 23rd International Conference on World Wide Web*, pp. 925–936. ACM (2014)
4. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: a complex systems look at the underlying process of word-of-mouth. *Mark. Lett.* **12**(3), 211–223 (2001)
5. Gomez Rodriguez, M., Leskovec, J., Schölkopf, B.: Structure and dynamics of information pathways in online media. In: *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013*, pp. 23–32. Association for Computing Machinery, New York (2013)
6. Ikeda, K., Sakaki, T., Toriumi, F., Kurihara, S.: Report of findings obtained from modeling of false rumor diffusion in case of disaster. In: *The 31st Annual Conference of the Japanese Society for Artificial Intelligent JSAI2017*, 3P1–NFC–00a–1 (2017)
7. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 137–146. ACM (2003)
8. Kim, J., Bae, J., Hastak, M.: Emergency information diffusion on online social media during storm Cindy in U.S. *Int. J. Inf. Manag.* **40**, 153–165 (2018)
9. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: *AAAI*, vol. 7, pp. 1371–1376 (2007)
10. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78**(4), 046110 (2008)
11. Li, M., Wang, X., Gao, K., Zhang, S.: A survey on information diffusion in online social networks: models and methods. *Information (Switzerland)* **8** (2017)
12. Murata, T., Koga, H.: Extended methods for influence maximization in dynamic networks. *Comput. Soc. Netw.* **5** (2018)
13. Osawa, S., Murata, T.: Selecting seed nodes for influence maximization in dynamic networks. *Stud. Comput. Intell.* **597**, 91–98 (2015)

14. Takashi, K., Masashi, T., Naoki, Y.: Detecting information cascades with social influence from microblogs. *Inf. Process. Soc. Jpn. Trans. Database* **9**(2), 23–33 (2016)
15. Watts, D.J.: A simple model of global cascades on random networks. *Proc. Natl. Acad. Sci.* **99**(9), 5766–5771 (2002)
16. Watts, D.J., Dodds, P.S.: Influentials, networks, and public opinion formation. *J. Consum. Res.* **34**(4), 441–458 (2007)
17. Yuya, Y., Kazumi, S., Hiroshi, M., Kouzou, O., Masahiro, K.: Estimating method of expected influence curve from single diffusion sequence on social networks. *IEICE Trans. Inf. Syst.* **94**(11), 1899–1908 (2011)



Diffusion Dynamics Prediction on Networks Using Sub-graph Motif Distribution

Alexey L. Zaykov, Danila A. Vaganov, and Valentina Y. Guleva^(✉)

ITMO University, 49 Kronverksky Pr., 197101 Saint-Petersburg, Russian Federation
guleva@itmo.ru

Abstract. Motifs are believed to represent structural and dynamical properties in networks. Nevertheless, small motifs are not always representative, while large motifs are hard to evaluate, which results in problem of recognition of optimal motif size, effective algorithms development, and motif significance estimation. We explore, in which extent diffusion dynamics on a graph can be estimated on the base of its subgraphs and motifs in particular. For this purpose we explore and compare motifs distributions for initial graph and its samples extracted by different techniques, and analyse how subgraph sizes affect prediction accuracy of diffusion time on the base on motifs. This allows to understand which subgraph sizes are appropriate for such kind of prediction, and how can we represent subgraph structural patterns to use smaller samples for dynamics approximations on large graphs. Several sampling techniques are compared for VK dataset with interest attribute markup. 4–5 node motifs are taken for graphs representation and for prediction evaluation.

Keywords: Network diffusion · Network motifs · Process spreading · Sub-graph sampling

1 Introduction

Finding key patterns in network structures explains the main semantic differences and network formation process. Motifs are structural units, characterising network systemic properties.

Different kinds of networks can be characterised by motifs and can be distinguished in this way. Therefore, motifs are often interpreted as functional units in biological, chemical, or other kinds of networks. They are associated with functional groups and are used to explore emerging functional patterns in dynamics [1]. In contrast to it, functional properties in social and economical groups are weakly studied, therefore, extracted motifs are hard to interpret or validate. In addition, interest networks, as a special type of social networks, are characterised by special “motifs”, peculiar to the special type of a networks and associated

dynamics. Functionally, they reflect how people connect each other in the context of their goals in interest societies, or how people are different from molecules at the global scale. In this way, there exists a global research question of how to extract these basic structural patterns, characterising a network specificity and reproducing its functional/dynamic properties.

Extraction of these patterns contributes to the estimation of dynamics on a whole network on the base of its sub-graph properties, which is of great importance for understanding dynamics of large networks of different types and for computational performance optimisation. Existing methods of motifs extraction allow for 6-node motifs extraction, which is not enough to distinguish some kinds of networks and to reflect their dynamical properties. Extraction of larger motifs is time-consuming. Therefore, in current study we compared sampling techniques for sub-graphs extraction and explored how diffusion dynamics on graphs can be predicted on the base of its sub-graph properties.

In order to select the most appropriate samples, we compared their motif structures with initial network. We assume, that too small sub-graph does not capture all motifs, while too large sub-graph may be equal to initial network or comprise required structural patterns multiple times. In this way, we have chosen a sampling technique, returning the smallest divergence between motif sequences, and then for each network we selected its smallest sub-graph, having similar motif sequence to an initial graph. To estimate prediction ability, we took motif sequences of sub-graphs as an input of regression model and explored how these motifs can predict SI diffusion dynamics on a whole network.

As a data set VK friendship networks with various topical attribution was explored. We extracted not intersected 4–5-node motifs by **SuperNoder** method and analysed motif significance, which was shown to contrast with results of **Gtries**, extracting intersected motifs. After that, we compared a number of sampling techniques, and explored how the obtained sub-graphs differ from initial graph in Kullback-Leibler divergence for motif distributions. Finally, we selected minimal sub-graph size, showing appropriate divergence. For those graphs diffusion dynamics time was predicted on the base of motif distribution.

2 Literature

Here we explore how motifs are connected with dynamics on networks, and how implementations of motif extraction algorithms differ.

2.1 Motifs and Dynamics on Networks

Motifs relation to dynamics is explored to build a connection with functional properties and to understand formation pattern better. This is also believed to allow to conclude dynamical and functional properties of a large network on the base of its small pattern. In this way, some studies, related to dynamics, are mostly focused on functional dynamic properties [2] demonstrated by single motives. In contrast, Ingram et al. [8] argue there can not be a strict connection

between a motif structure and its function: authors explored dynamical responses of motifs, having “bi-fan” network structure and differed by node properties, in biological networks. This can be due to sizes of chosen motifs, absence of difference of given networks, and so on. Function of motifs, observed in social networks, may be explained by social restrictions like Dunbar’s number [6], at the same time interaction patterns in interest networks, depending on personal or group goals are not enough formalised for motifs interpretation.

Estimation of synchronisation dynamics in large networks on the base of motifs [11] is provided by means of eigenvectors [20] of connection matrices, obtained from initial motif by Kroneker product. Lodato et al. [15] explore synchronisation dynamics for 3 and 4-node motifs, and analyse which of them are correlated with stability states. D’Huys et al. [4] study Kuramoto oscillation models for three kinds of network motifs with different symmetries, and find numerical solutions for those single motifs. Nevertheless, this result is not expanded on the whole network. Motives synchronisation is also discussed in [33].

Contribution of motives to dynamics on networks is studied by [21], showing how motif abundance affect structural stability score. They show the significance of combination of motifs, having particular structural properties, with their frequency. In this way, the explored networks are divided into groups according to aggregated properties, which is observed for both, 3 and 4-node motives.

[31] explore diffusion at individual and population scales in relation to motif structure and try to infer diffusion network with motif profile. Finally, diffusion networks are considered. Sarkar et al. [28] also use motifs to understand whether or not they can be used for explanation of emerging cascades. For this purpose edges, covered by percolation algorithm are compared with edges generated by motifs. In this way, existing studies related to diffusion dynamics are mostly focused on dynamic paths generated, and distinguish structural and process-related motifs [29]. The majority of methods cover small motifs due to high algorithmic complexity of this procedure and dynamics on whole networks is restricted by diffusion paths.

In this study we explore, in which extent diffusion dynamics on a graph can be estimated on the base of its subgraph. For this purpose we explore and compare motifs distributions for initial graph and its samples, and analyse how subgraph sizes affect prediction accuracy on the base on motifs. This extends question of diffusion dynamics estimation with motifs, on one side, but on other, this allows to understand which subgraph sizes are appropriate for such kind of prediction, and how can we represent subgraph structural patterns to use small samples for dynamics approximations on large graphs.

2.2 Motif Detection Methods Implementations

The vast majority of existing algorithms was covered in [22]. Existing implementations, concerning questions of efficiency [34], allow to process 1000 graphs for 4–5 node motives for 8 min. In addition, there are implementations for approximate subgraph counting, i.e. RAND-Gtrie [23] and RAND-FaSE [19], which have no restriction to the size of motif count. SuperNoder [9] allows to extract

Table 1. Comparison of implementations efficiency and motifs quality

	k -restriction	z-scores	Parallel	Time
ESU [35]	None	✗	✗	21.624
Kavosh [10]	None	✓	✗	10.487
FaSE [18]	None	✓	✓	0.877
SubEnum [30]	None	✗	✓	2.726
gtrieScanner [23, 25, 26]	≤ 50	✓	✓	0.322

non-intersected motifs by sub-graphs embedding and reproduce self-similarity of networks, nevertheless, this increases computational time (Table 1).

3 Network Data

As the main data the set of friendship graphs was used. Nodes represent users, subscribing communities in VK social network, and edges between them reflect friendship. Neither nodes nor edges have additional attributes. In total we had 418 graphs with interest attribute markup. Topics were marked up by expert to provide balance between group sizes, and contain reach variability in topological properties (the statistical analysis of topology for the data set can be found in [32]).

4 Method

4.1 Motif Detection Methods

Motifs extracted were very desired to contain enough nodes to represent graph structural patters, responsible for process spreading, and to split graph into disjunctive components. In this way, methods for motifs extraction were selected according to their computational performance and edges intersection in graph partitions.

First, motif approximations were obtained by building a prefix tree (g-trie method [24]) for each node in a base graph. This method effectively evaluates motif of 5 nodes, nevertheless, the obtained samples do not split graph into disjunctive structural components.

As a non-intersected motif extraction method, the **SuperNoder** algorithm was explored [3]. It decomposes a network after each iteration by folding motifs into a node. In this way, self-similar patterns can be extracted, but computation efficiency decreases. As a result, the distribution of the embedded motifs and the rest of the nodes are calculated.

Motifs significance was evaluated by z-scores [17]. They were combined with the corresponding frequencies and used as input features in further classification tasks.

4.2 Subgraph Sampling for Representation of Motifs

The task of motif calculation is exhaustive, thus the variety of methods is impossible to apply for large networks. In this case, we assume that local structure of different parts in the real-world network emerged under the same process. Therefore, one needs only a certain part of a graph to perform precise analysis. In this way, we aimed at choosing the best sampling strategy, which allows for correct reflection of a graph properties via its sub-graph of minimal possible size, but representing main structural particularities.

To estimate representativity and ability of sample to reproduce functional properties of initial graph, we go back to motifs distribution. With example of modular graph it is easy to see, that if a sample is small, it comprises only motifs within a module, while a larger sample will take both kinds of motifs, within and between modules. Finally, too large motifs would be comparable with a whole graph by size, which have no sense, despite motif distribution similarity. In this way, we calculate Kullback–Leibler (KL) divergence between motif distributions (defined in previous section) of a sampled sub-graph and an original graph, and try to select minimal sub-graph with minimal KL divergence as a sample.

For implementation of different sampling strategies the library `Little Ball of Fur` [27] was used. To evaluate the motif representativity of sub-graphs, we compare the following sampling strategies:

- Common Neighbor Aware Random Walk Sampler [14]
- Metropolis Hastings Random Walk Sampler [7]
- PageRank Based Sampler [13]
- Random Walk Sampler [5]
- Community Structure Expansion Sampler [16]
- Breadth First Search Sampler [12]

We use sample-strategies, allowing to control a number of nodes in a resulting sub-graph, hence we vary the fraction of a sub-graph by 10% and repeat calculation of divergence with a different random seeds 5 times for each graph considered.

4.3 Motifs and Process Spreading: Regression Task Statement

After motif extraction step we obtain a sequence of relative frequencies for each extracted motif of a given network. These frequencies form a vector, coming as an input to a regression model, which aims at diffusion time prediction. Let diffusion time be a number of iterations for system to come from a state with all not activated nodes to all activated. The diffusion process is discrete and implemented as Susceptible-Infected epidemic model on a graph, where activated nodes may infect susceptible ones with a given probability if they are connected by an edge of a graph. In this way, each iteration corresponds to going over all infected nodes and infect their neighbours with a given probability.

Prediction quality of the obtained regression model is estimated by MAPE feature and significant motifs, affecting model performance, are evaluated by

SHAP values. Data partitioning for train and test sets is made by cross-validation technique.

4.4 Self-similarity and Dynamics on Networks

Finally, we explore self-similar properties of graphs in terms of their motif distribution. We consider graphs of different sizes and sample subgraphs by different methods. Then we compared the obtained motifs distributions of samples with initial graph motif distributions. We explored this result for different sizes of graph in the combination with sample sizes, and explored the connection with Kulback-Leibler divergence. Then we took transitional sample size, which demonstrated minimal divergence in the combination with percentage of sample. In addition, for each graph diffusion times were known. In this way self-similar samples, in terms of motif distribution, were detected.

The second issue was to work with dynamics and motifs. For the data set obtained we had a sample, motif distribution for the sample, motif distribution for a graph, and diffusion time for whole network explored to density. In this way, We took data with small divergence values and try to estimate diffusion time for networks on the base of sample properties. In this way, we compared regression quality with KL divergence.

5 Results

5.1 Dynamics on Networks and Motifs

To connect diffusion dynamics with motifs, we considered regression task for networks, where frequencies of motifs are input graph features, and number of iteration for diffusion over the whole network is a predicted value.

Dynamics was simulated by susceptible-infected epidemic model, and time, taking diffusion process, was considered as a predicted value. **Z-score** was evaluated as a significance measure for extracted motif and they were taken as predictors in the combination with frequencies. Motifs were evaluated with **G-trie** method and contained 4–5 nodes.

The most dense and sparse structures affected prediction for frequencies, in the case of certain feature values (f-22, f-16). At the same time, z-8 and z-14 have lower density, in this way, their z-values are not contrasted by densities of certain networks, in this way, their significance to prediction seems to be higher (Fig. 1).

Higher mean prediction error and its deviation was demonstrated for networks with the highest density since it corresponds to the smallest networks. In addition, this is the widest bin. Number of nodes is not significant for prediction accuracy, since variation of network sizes is valuable for densities up to 0.0209. Networks with less than hundred nodes and 0.2 density show the highest error (Fig. 2).

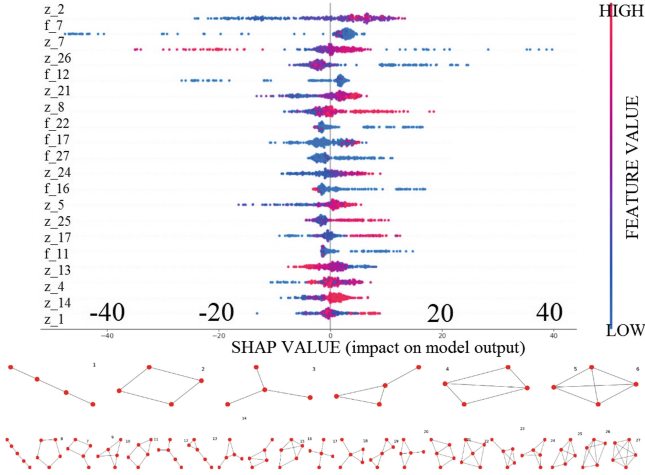


Fig. 1. Impact of motifs and their relative frequencies on diffusion dynamics prediction power. The motifs with the highest impact are displayed in decreasing order. Points correspond to graphs, their colour show relative frequency of the motif in the graph

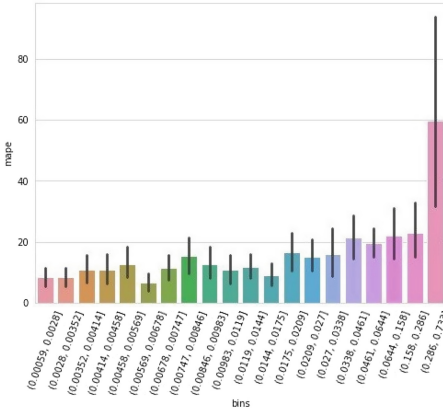
5.2 Sampling Techniques

To find better subgraph for estimation of global dynamics we do sampling. Sampling is similar to motif extraction task, nevertheless, it takes other scale. In order to understand which subgraph sizes and their structural properties better match regression task needs, we compare different sample methods on a full data set, and measure Kullback-Leibler divergence between sampled and original motif distributions as a performance measure (Fig. 3 left). The lowest value of mean KL on VK data set is shown by Community Structure Expansion Sampler (which was used further in experiments). Methods performance was shown to depend on relative sample size, nevertheless, Expansion Sampler performed others upto 0.6 sample relative size (Fig. 3 right).

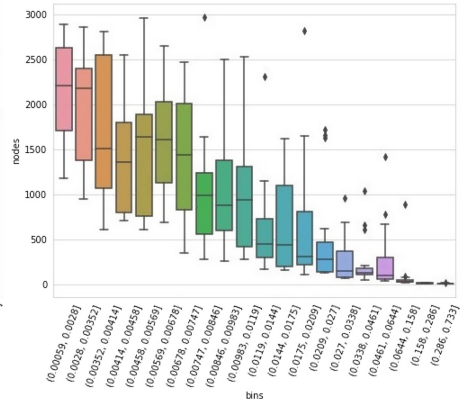
5.3 Self-similarity and Dynamics

Motif distributions were compared for subgraphs of different sizes, and then prediction ability of initial graph dynamical properties on the base of subgraphs was explored (Fig. 4).

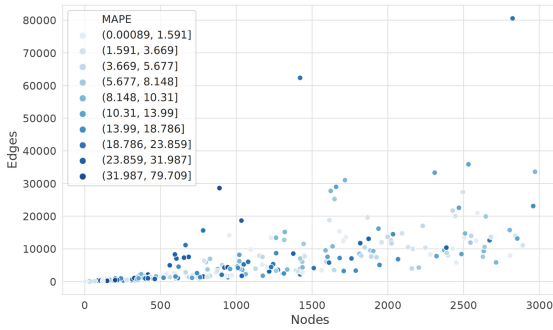
Increase in sub-graph relative size naturally lead to decrease in divergence between motifs distribution, since taking of 100% sub-graph equals to initial graph. Averaging over all graphs show sharp decrease of divergence upto 185 nodes, after which a small hop is observed and then decrease is continued. For this reason we select graphs with more than 1000 nodes and sample sub-graphs of 185 nodes, and predict dynamics for whole graphs on the base of sub-graph properties. For comparison, we display the dependence of prediction quality on KL divergence between sub-graph and graph motifs (Fig. 5).



(a) Error against density



(b) Number of nodes against density



(c) MAPE against nodes and edges

Fig. 2. Connection between prediction accuracy and network sizes and density

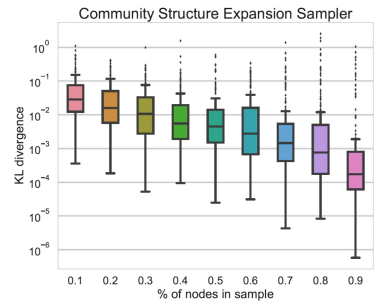
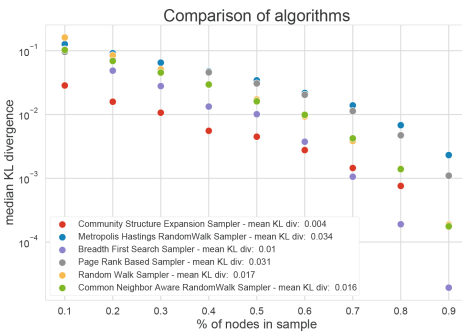
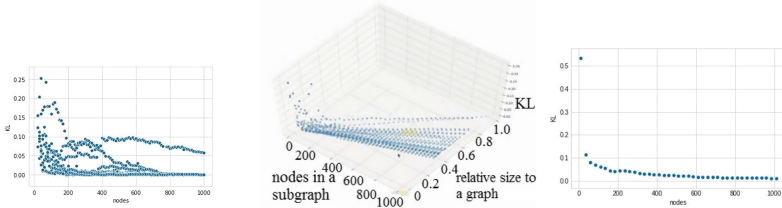


Fig. 3. Dependence of KL divergence on sample relative sizes for different sampling methods (left) and for the best method (right)



(a) Divergence against sub-graph sizes (b) Sub-graph size and relative size (c) Averaged divergence against sub-graph size

Fig. 4. Kullback-Leibler divergence between motif distributions for graphs and sub-graphs of different relative sizes

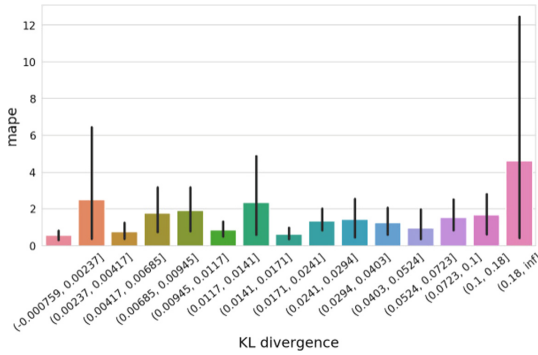


Fig. 5. Diffusion speeds prediction accuracy depending on divergence between sub-graph and graph motif distribution

Results show divergence does not really affect prediction quality significantly, which indicate the considered relative size of samples is appropriate for estimations. Nevertheless, KL divergence after 0.18 demonstrate clear increase in deviance and mean error. This means, these types of networks require additional analysis of their structural patterns and motifs inside.

6 Discussion and Conclusion

This study explores structural properties of interest networks in terms of motifs. We study, if diffusion dynamics on networks can be estimated on the base of its building blocks. For this purpose we sample sub-graph by Community Structure Expansion Sampler, providing the best similarity between sample and initial graphs in terms of their motif sequences. Then we use motif frequencies of the sample as input features for regression model, and estimate diffusion dynamics for the whole graph. Diffusion dynamics is estimated as number of iterations before all nodes activation and is considered as an output in the regression task. Results show sampling method approximate initial graph more accurate

for larger percentage of nodes in a sample. In addition, KL divergence for motifs distributions follow the similar law. Prediction error is not related to the network size, but increasing error is associated with increasing density, since this corresponds to rare small networks with high density. Divergence between motif distributions show weak affect on prediction error before 0.18 value. At the same time, the consideration of smaller sub-graphs seems to significantly enrich future studies due to variety of densities and increasing variability of KL divergence. Estimation of dynamics at sample level seems to be possible, nevertheless, future studies are aimed at investigation of these samples structural properties and developing methods of their concatenation to obtain initial networks.

Acknowledgements. This research is financially supported by The Russian Science Foundation, Agreement # 19-71-00153.

References

1. Alexander, R.P., Kim, P.M., Emonet, T., Gerstein, M.B.: Understanding modularity in molecular networks requires dynamics. *Sci. Signal.* **2**(81), pe44–pe44 (2009)
2. Alon, U.: Network motifs: theory and experimental approaches. *Nat. Rev. Genet.* **8**(6), 450–461 (2007)
3. Dessì, D., Cirrone, J., Recupero, D.R., Shasha, D.: SuperNoder: a tool to discover over-represented modular structures in networks. *BMC Bioinform.* **19**(1), 318 (2018)
4. D’Huys, O., Vicente, R., Erneux, T., Danckaert, J., Fischer, I.: Synchronization properties of network motifs: influence of coupling delay and symmetry. *Chaos: Interdiscip. J. Nonlinear Sci.* **18**(3), 037116 (2008)
5. Gjoka, M., Kurant, M., Butts, C.T., Markopoulou, A.: Walking in Facebook: a case study of unbiased sampling of OSNs. In: 2010 Proceedings IEEE Infocom, pp. 1–9. IEEE (2010)
6. Gonçalves, B., Perra, N., Vespignani, A.: Modeling users’ activity on twitter networks: validation of Dunbar’s number. *PLoS ONE* **6**(8), e22656 (2011)
7. Hübler, C., Kriegel, H.P., Borgwardt, K., Ghahramani, Z.: Metropolis algorithms for representative subgraph sampling. In: 2008 Eighth IEEE International Conference on Data Mining, pp. 283–292. IEEE (2008)
8. Ingram, P.J., Stumpf, M.P., Stark, J.: Network motifs: structure does not determine function. *BMC Genom.* **7**(1), 1–12 (2006)
9. Irigoien, F., Triolet, R.: SuperNode partitioning. In: Proceedings of the 15th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pp. 319–329 (1988)
10. Kashani, Z.R.M., Ahrabian, H., Elahi, E., Nowzari-Dalini, A., Ansari, E.S., Asadi, S., Mohammadi, S., Schreiber, F., Masoudi-Nejad, A.: Kavosh: a new algorithm for finding network motifs. *BMC Bioinform.* **10**(1), 1–12 (2009)
11. Krishnagopal, S., Lehnert, J., Poel, W., Zakharova, A., Schöll, E.: Synchronization patterns: from network motifs to hierarchical networks. *Philos. Trans. R. Soc. A: Math. Phys. Eng. Sci.* **375**(2088), 20160216 (2017)
12. Lee, C.H., Xu, X., Eun, D.Y.: Beyond random walk and Metropolis-Hastings samplers: why you should not backtrack for unbiased graph sampling. *ACM SIGMETRICS Perform. Eval. Rev.* **40**(1), 319–330 (2012)

13. Leskovec, J., Faloutsos, C.: Sampling from large graphs. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 631–636 (2006)
14. Li, Y., Wu, Z., Lin, S., Xie, H., Lv, M., Xu, Y., Lui, J.C.: Walking with perception: efficient random walk sampling via common neighbor awareness. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE), pp. 962–973. IEEE (2019)
15. Lodato, I., Boccaletti, S., Latora, V.: Synchronization properties of network motifs. *EPL (Europhys. Lett.)* **78**(2), 28001 (2007)
16. Maiya, A.S., Berger-Wolf, T.Y.: Sampling community structure. In: Proceedings of the 19th International Conference on World Wide Web, pp. 701–710 (2010)
17. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: simple building blocks of complex networks. *Science* **298**(5594), 824–827 (2002)
18. Paredes, P., Ribeiro, P.: Towards a faster network-centric subgraph census. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 264–271 (2013)
19. Paredes, P., Ribeiro, P.: Rand-FaSE: fast approximate subgraph census. *Soc. Netw. Anal. Min.* **5**(1), 17 (2015)
20. Poel, W., Zakharova, A., Schöll, E.: Partial synchronization and partial amplitude death in mesoscale network motifs. *Phys. Rev. E* **91**(2), 022915 (2015)
21. Prill, R.J., Iglesias, P.A., Levchenko, A.: Dynamic properties of network motifs contribute to biological network organization. *PLoS Biol.* **3**(11), e343 (2005)
22. Ribeiro, P., Paredes, P., Silva, M.E., Aparicio, D., Silva, F.: A survey on subgraph counting: concepts, algorithms and applications to network motifs and graphlets. arXiv preprint [arXiv:1910.13011](https://arxiv.org/abs/1910.13011) (2019)
23. Ribeiro, P., Silva, F.: Efficient subgraph frequency estimation with g-tries. In: International Workshop on Algorithms in Bioinformatics, pp. 238–249. Springer (2010)
24. Ribeiro, P., Silva, F.: G-tries: a data structure for storing and finding subgraphs. *Data Min. Knowl. Disc.* **28**(2), 337–377 (2014)
25. Ribeiro, P., Silva, F., Lopes, L.: A parallel algorithm for counting subgraphs in complex networks. In: International Joint Conference on Biomedical Engineering Systems and Technologies, pp. 380–393. Springer (2010)
26. Ribeiro, P., Silva, F., Lopes, L.: Parallel discovery of network motifs. *J. Parallel Distrib. Comput.* **72**(2), 144–154 (2012)
27. Rozemberczki, B., Kiss, O., Sarkar, R.: Little ball of fur: a python library for graph sampling. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM 2020). ACM (2020)
28. Sarkar, S., Guo, R., Shakarian, P.: Using network motifs to characterize temporal network evolution leading to diffusion inhibition. *Soc. Netw. Anal. Min.* **9**(1), 14 (2019)
29. Schwarze, A.C., Porter, M.A.: Motifs for processes on networks. arXiv preprint [arXiv:2007.07447](https://arxiv.org/abs/2007.07447) (2020)
30. Shahrivari, S., Jalili, S.: Fast parallel all-subgraph enumeration using multicore machines. *Sci. Program.* **2015** (2015)
31. Tan, Q., Liu, Y., Liu, J.: Motif-aware diffusion network inference. *Int. J. Data Sci. Anal.* **9**(4), 375–387 (2020)
32. Vaganov, D.A., Guleva, V.Y., Bochenina, K.O.: Social media group structure and its goals: building an order. In: International Conference on Complex Networks and their Applications, pp. 473–483. Springer (2018)

33. Vega, Y.M., Vázquez-Prada, M., Pacheco, A.F.: Fitness for synchronization of network motifs. *Physica A* **343**, 279–287 (2004)
34. Wernicke, S.: Efficient detection of network motifs. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **3**(4), 347–359 (2006)
35. Wernicke, S., Rasche, F.: FANMOD: a tool for fast network motif detection. *Bioinformatics* **22**(9), 1152–1153 (2006)



Using Distributed Risk Maps by Consensus as a Complement to Contact Tracing Apps

Miguel Rebollo^{1,2(✉)}, Rosa M. Benito², Juan C. Losada², and Javier Galeano²

¹ VRAIN - Valencian Research Institute for Artificial Intelligence,
Universitat Politècnica de València, Camino de Vera S/N 46022, Valencia, Spain
mrebollo@vrain.upv.es

² Complex Systems Group, Universidad Politécnica de Madrid,
c/ Ramiro de Maeztu, 7, 28040 Madrid, Spain
{[rosa.benito](mailto:rosa.benito@upm.es),[juancarlos.losada](mailto:juancarlos.losada@upm.es),[javier.galeano](mailto:javier.galeano@upm.es)}@upm.es

Abstract. The rapid spread of COVID-19 has demonstrated the need for accurate information to contain its diffusion. Technological solutions are a complement that can help citizens to be informed about the risk in their environment. Although measures such as contact traceability have been successful in some countries, their use raises society's resistance. This paper proposes a variation of the consensus processes in directed networks to create a risk map of a determined area. The process shares information with trusted contacts: people we would notify in the case of being infected. When the process converges, each participant would have obtained the risk map for the selected zone.

A consensus simulation has been introduced in an SEIR model to evaluate how having available a risk map could affect the virus's propagation. The scenario chosen is La Gomera Island: a region where the Spanish government has tested its contact tracing app (RadarCOVID). The paper also compares both strategies joint and separately: contact tracing to detect potential infections, and risk maps to avoid movements into conflictive areas. Contact tracing apps could work with 40% of participants instead of 60%. On the other hand, the elaboration of risk maps could work with just a 20% of active installations. Nevertheless, the effect is to delay the propagation instead of reducing the contagion. With both strategies actives, we significantly reduce infected peoples with a relatively low participant number.

Keywords: Consensus · Complex network · COVID · Risk map · Collaboration · Contact tracing

1 Introduction

One of the current challenges to stem the spread of COVID-19 is to track people infected with coronaviruses that can spread the disease. Although technological

solutions such as contact traceability have been successful in some countries, they raise resistance in society due to privacy concerns [15]. The European Data Protection Board has published a guideline for the governments to use this kind of technology, guaranteeing privacy, and proper access to the data [13]. The solutions currently into consideration fall into two main groups: (1) personalized tracking of users from its geolocation, and (2) private tracking of contacts. Most governments have recommended using the second type of application, advising against proposals based on individuals' geolocation.

Simko et al. [12] made a series of surveys over 100 participants to analyze their opinion about contact-tracking applications and privacy. It is a relevant study since the first part finished when some European countries were under different forms of lock-downs, and contact-tracing apps were not available yet. Between the first and the second study, several proposals appeared, such as the ones made by Apple and Google [1,2], the Massachusetts Institute of Technology (MIT) [10], the University of Washington (UW) [4], PEPP-PT [9], Inria [3], WeTrace [16], and DP3T [14]. The study throws that people are more comfortable using an existing mapping application that adds tracking for COVID-19 instead of using new apps, with reservations even if they provide 'perfect' privacy. One of the main concerns is sharing data, preferring that Google or organizations such as the UN develop the application. In general, participants manifest a lack of trust in how their governments would use the citizens' location data. They thought that it was unlikely that their government would erase the data after the crisis and also that they would use it for other purposes. For both studies, something in common was mixed feelings about using proximity tracking for the contacts and negative towards using any other data source.

The MIT Technology Review [8] has been collecting the different proposals that states have created. Currently, there are 43 registered apps. There are initiatives in the five continents, but most of the countries belong to Asia and Europe since they were the firsts places where COVID-19 appeared. The population that uses the applications varies from 9,000 inhabitants in Cyprus to 100,000,000 in India. The median value is 1,613,500. The Bluetooth technology is the solution that most countries have chosen, with 72% of the apps. Moreover, almost half of them use the API provided by Google and Apple. Despite the recommendation to avoid location services, 36% of the apps still use it.

Despite the efforts to develop technological solutions to track the propagation of COVID-19, the usage of the apps is not extended enough. That is why we propose a third method: a process of **dissemination in local environments**. The method exchanges information with known and trusted contacts only. The consensus for COVID (C4C) method is a variation of the consensus processes proposed by Olfati-Saber and Murray [7]. It is a dissemination process that allows a distributed calculation of the value of a function in a network, exchanging information only with direct neighbors without having global knowledge of the structure, size, values, or other characteristics of the graph. This process converges to a final single value for the calculated function. With this approach, privacy is maintained, the administration obtains aggregated information, and

citizens and the administration have the same data, promoting transparency. One relevant limitation is that some critical mass is still necessary.

The rest of the paper is structured as follows. Section 2 explains how citizens can collaboratively create risk maps using a consensus process with their close contacts. Section 3 shows the results using La Gomera as an example: one of the Canary Islands, with 21,550 inhabitants. The Spanish government carried out a pilot project with their contact tracing application (RadarCOVID) on that island, so we decided to use the same scenario. Finally, Sect. 4 summarizes the main findings of this work.

2 Collaborative Risk Map Generation

The consensus for COVID (C4C) proposal works over a contact network with non-reciprocal relationships. It is needed to avoid the presence of hubs with an elevate number of contacts. Therefore, the underlying structure is a directed graph. The original consensus algorithm works over non-directed graphs, so we have to extend the model to consider this case.

2.1 Extension of the Consensus Process

Let $G = (V, E)$ be a non-directed network formed by a set of vertices V and a set of links $E \subseteq V \times V$ where $(i, j) \in E$ if there is a link between the nodes i and j . We denote by N_i the set formed by the neighbors of i . A vector $x = (x_1, \dots, x_n)^T$ contains the initial values of the variables associated with each node. Olfati-Saber and Murray [7] propose an algorithm whose iterative application converges to the mean value of x .

$$x_i(t + 1) = x_i(t) + \varepsilon \sum_{j \in N_i} [x_j(t) - x_i(t)] \tag{1}$$

The authors demonstrated that this consensus process converges to the average of the initial values when $\varepsilon < \frac{1}{\max d_i}$, being d_i the degree of node i . There is an equivalent matricial formulation.

$$x(t + 1) = \underbrace{(I - \varepsilon L)}_P x(t), \quad \text{with } L = D_{A_G} - A_G \tag{2}$$

where I denotes the identity matrix and L is the laplacian of the adjacency matrix of the graph G . This expression P is called the Perron–Frobenius matrix and governs the consensus process’s collective dynamics.

If each component contains a vector $x_i = (x_i^1, \dots, x_i^m) \in \mathbb{R}^m$, the process carries out a consensus over m independent variables. By expanding the vector with one additional element $y_i \in \mathbb{R}$, we can determine the size of the network at the same times as follows: $x_i \oplus y_i = (x_i^1, \dots, x_i^m, y_i)$. Initially, $y_i = 0 \forall i$. Without losing generality, we can introduce an additional node in the network

whose initial values are $x_0 \oplus y_0 = \underbrace{(0, \dots, 0)}_m, 1$). As the process converges to the average value, $y_i = 1/|V|$ and, therefore, $|V| = 1/y_i$ is the size of the network.

We need the Perron P matrix to be doubly stochastic for the consensus to work: a matrix whose rows and columns add up to one. However, in directed networks, we obtain a row stochastic one. Inspired in the Dominguez-Garcia and Hadjicostis matrix scaling algorithm [5], we define an iterative process to convert the Perron matrix into a double stochastic one. The process begins with a row stochastic matrix \bar{P} , and, in each iteration, the matrix scales following the expression.

$$P(t) = \bar{P}\Delta(t) + [I - \Delta(t)] \quad (3)$$

where \bar{P} is a local Perron matrix and $\Delta(t)$ is updated as Algorithm 1 describes. The only consideration is that the Perron matrix \bar{P} is defined using the degree of each node instead of a common ε value for all the nodes (see line 4). Furthermore, as $P(t)$ is based on the Perron matrix, we can combine the matrix's scaling with the consensus value calculation in the same step (line 12).

Algorithm 1. Matrix scaling and consensus (collective)

```

1: init  $x(0)$ 
2:  $L = D_A^{out} - A$ 
3:  $\Delta(0) = D_L^{-1}$ 
4:  $\bar{P} = (I - \Delta(0)) * L$ 
5:  $P(0) = \bar{P}\Delta(0) + [I - \Delta(0)]$ 
6:  $\pi(0) = \mathbf{0}, \quad \eta(0) = \mathbf{1}$ 
7: repeat
8:    $\pi(t+1) = P(0) \pi(t)$ 
9:    $\eta(t+1) = \max(\pi(t), \max \eta(t)A)$ 
10:   $\Delta(t+1) = \Delta(0) \frac{\pi(t+1)}{\eta(t+1)}$ 
11:   $P(t+1) = \bar{P}\Delta(t+1) + [I - \Delta(t+1)]$ 
12:   $x(t+1) = P(t+1) x(t)$ 
13: until  $x(t)$  converges
    
```

The adaptations of the scaling algorithm are the calculation of $\Delta(t)$ (lines 3 and 10), the definition of the matrix \bar{P} as a local variation of the Perron matrix (line 4), and how $\pi(t)$ updates (line 8).

2.2 Map Generation

Once we have defined the algorithm for consensus processes over directed networks, the goal is to create a citizen network in an area (town, province, state, or any other administrative division) that uses it to collaboratively create a risk map. We have chosen the census districts from the National Institute of Statistics (INE) of Spain. The sizes of the districts are relatively homogeneous, having between 900 and 3,000 inhabitants each one. It is easily scalable, aggregating

Algorithm 2. Risk map creation

- 1: calculate ri_i
 - 2: init $x_i(0) \oplus y_i(0) = (0, \dots, ri_i, \dots, 0, 0)$,
 - 3: execute Algorithm 1 until convergence
 - 4: calculate $R_i = \frac{x_i(t)}{y_i(t)}$
-

the information in bigger administrative units. Moreover, they never provide statistics with less than 100 persons to avoid reidentification.

Inhabitants share a risk index (RI) that measures their probability of being infected by COVID-19. The risk in a census district depends on the RI of all the people that live in it. The RI could integrate data from different sources: medical symptoms, symptoms from the close contacts, age, family situation, or habitability conditions. In this work, we use the same measure as the emergency service 112.¹ The risk value depends on medical symptoms: shortness of breath (60 points), fever (15), coughing (15), or close contact (29). Over 30 points, it is considered that the person has been infected. We use the following notation:

- ri_i : risk index of node i , $i = 1, \dots, n$
- $x_i \oplus y_i = (x_i^1 \dots, x_i^m, y_i)$: vector with the risk map values in node i .
- $R_i = (ri_1, \dots, ri_n)$: complete risk map

Let us assume an extra node representing an administrative unit, such as the town hall, acts as the x_0 node. Algorithm 2 describes the complete process.

Some important remarks related to the process are:

1. the position of ri_i in $x_i(0)$ corresponds to its census district
2. each node executes a local version of Algorithm 1
3. the first exchange is the only moment in which vectors contain individual values: the risk and the census district of i . We assume that there are no privacy concerns since the node would share this information with its N_i
4. in the following exchanges, the vectors received $x_j(t)$ contains aggregated information. As the neighbors of j remain unknown for i , it is not easy to track back the data.

It is a successive refinement mechanism: there is a map available at any time, and the longer the algorithm executes, the fittest the risk values are (see Fig. 4). The final risk map R is the same one that a centralized process would obtain with all the risk indexes available.

3 Results

The purpose of this section is to validate the algorithm proposed to create risk maps in a scenario similar to the conditions of the real world. The ideal situation would be to launch a pilot project in a controlled environment. However,

¹ <https://coronavirus.comunidad.madrid/>.

we consider that a previous simulation is essential. Therefore, the population model, the mobility patterns, and the SEIR models are defined to provide an environment with the characteristics relevant for the algorithm.

3.1 Population and Infection Model

As an application example, we have chosen La Gomera: one of the Canary Islands with 21,550 residents. The National Institute of Statistics divides the island into 14 census districts. The population that lives in each area is publicly available. The network has as many nodes as inhabitants. For each node, we generate the coordinates for their home address (Fig. 1, left). They are random coordinates following the density distribution of the census districts.

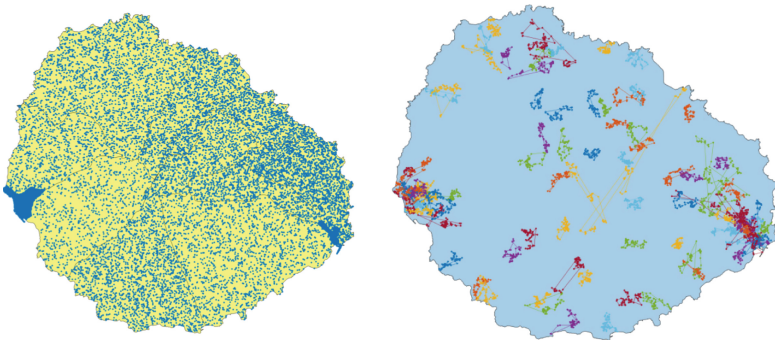


Fig. 1. (Left) Population distribution in La Gomera. (Right) Sample of 100 paths using Lévy flights

The movements of the people along the day are simulated using recurrent Lévy flights [6]. Each person has assigned a path with 96 points (every 5 min for 8 h) that begins and ends at his or her home location (Fig. 1, right). We have validated the model comparing the movements with the data available in the study on mobility based on mobile phone carried out by the Spanish National Statistics Institute (INE) in 2019.² In this study, La Gomera was divided into two areas. The flows represent travels from commuters between both areas. No external sources, such as ferry or plane trips from other islands or the peninsula, would undoubtedly be relevant. This data, if available, could be added to the model. The daily mobility between them was 450 persons leaving San Sebastián de la Gomera area and 550 enter (the inverse from the northern area viewpoint). The simulation with Lévy flights throws an output flow of 464 persons and an input flow of 668. The movements are in the same magnitude order, so we assume that they are coherent.

² https://www.ine.es/en/experimental/movilidad/experimental_em_en.htm.

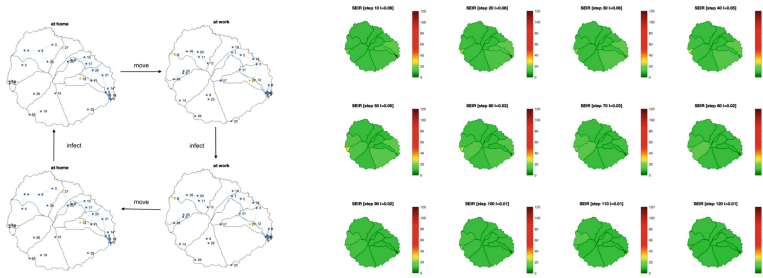


Fig. 2. (Left) The population moves alternatively between the home and the working location. Carriers can infect other people in both places. The cycle consists of a sequence movement → infection → movement → infection. (Right) Evolution of the risk map

To simulate the close contacts, we use the same criteria as the contact tracing app: a close contact is defined when two persons are at 5 meters (with 2 meters only obtains a 78% accuracy) at most and during 15 min. The result is a daily risky contact matrix of dimension 21,550 × 21,550.

Finally, to simulate the spread of the COVID-19, we use an SEIR model. Its parameters follow the findings from the literature that has analyzed the COVID-19 propagation [17]. Particularly, the incubation time is 7 days, so $\beta = 1/7$, the probability of infection $\sigma = 0.1$ and the recovery time is 15 days, so $\gamma = 1/15$. The purpose of the model is not to predict precisely the behavior of the disease. The model provides the consensus process with different scenarios to check the accurateness of the risk maps.

People start at their home location. They move along the day, interacting with the other persons. Nodes update their state according to the epidemic model and the contact matrix, and they go back to their home locations. A new infection stage is performed at home since, in COVID-19, some researches demonstrate the family to be a strong transmission source. Once completed the update, a new cycle begins (Fig. 2).

3.2 Risk Map Creation

The consensus process described in Algorithm 2 obtains the actual risk map if all the inhabitants participate in the process. However, as we have seen with contact tracing apps, this is a utopic scenario. Therefore, we assume that just 3,000 persons participate in creating the risk map (same volume as in the RadarCOVID contact tracing app).

To create the contact network, we have analyzed the network formed by the followers of the Twitter account of the town halls of cities of similar sizes. The degree of those networks follows a power-law distribution of parameter $\alpha \approx -1.7$. To model the contact network of the entire population, we have generated a preferential attachment network following the same distribution (Fig. 3, left). The resultant network had 58,000 contacts and a maximum degree of 876. As a potential application would bound the number of closer contacts, we choose a

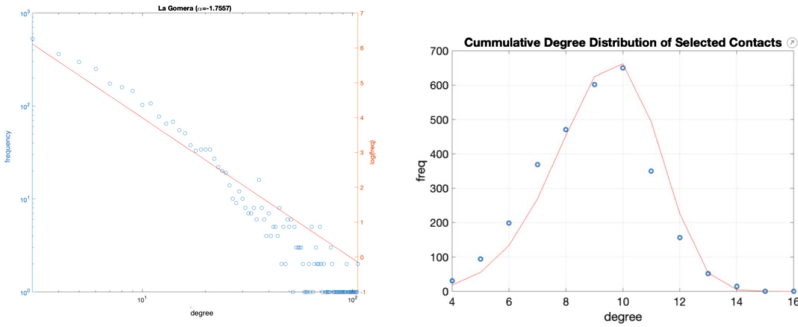


Fig. 3. Cumulated degree distribution of the networks (Left) Complete contact network with a power-law with $\alpha = -1.7$ (Right) Random selection of contacts. It follows a Weibull distribution with parameters $\alpha = 13.4$, $\gamma = 0.48$.

subset of the potential links. A reasonable limit is 15 contacts, five of each type (family, colleagues, and friends). If each person choose randomly 9 ± 2 contacts, the resulting network has 26,500 contacts and the number of connections vary from 4 to 16 (Fig. 3, right).

Therefore, we have obtained a network with 3,000 nodes and mean degree 10, varying from 4 to 16, generated from social network profiles. Over this scenario, the inhabitants can determine their town’s risk map at the end of the day. We assume that no additional measures, such as social distancing or limitations of movements, are taken.

As an example, let us consider the situation after 30 days. People have moved during this period as described in Sect. 3.1, and the contagion has evolved following the SEIR model. After 30 days, the situation of the COVID-19 in La Gomera appears in Fig. 4, with a clear breakout in San Sebastian de La Gomera (in red). The 3,000 persons determine their risk index (some of them are already infected) and share the value with their direct contacts, following the consensus process from C4C.

Each node has a vector of 14 components, one for each census district $x_i \oplus y_i = (x_i^1, \dots, x_i^{14}, 0)$. Let be ri_i the risk index of i and $cd_i = k$ the census district i lives in. $x_i^k = ri_i$ and the rest $x_i^l = 0$, $l \neq k$. Each node executes the process detailed in Algorithm 2. The evolution appears in Fig. 4. The real risk values are and the values calculated by consensus in one of the 3,000 participants for each census district are

Area	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Real risk	9.9	10.1	9.9	9.9	12.2	11.3	50.3	55.6	53.1	15.4	21.8	9.9	10.9	10.1
R_i	10.0	8.1	12.3	8.6	13.2	9.5	49.2	48.3	40.1	12.5	22.0	9.6	11.4	8.8

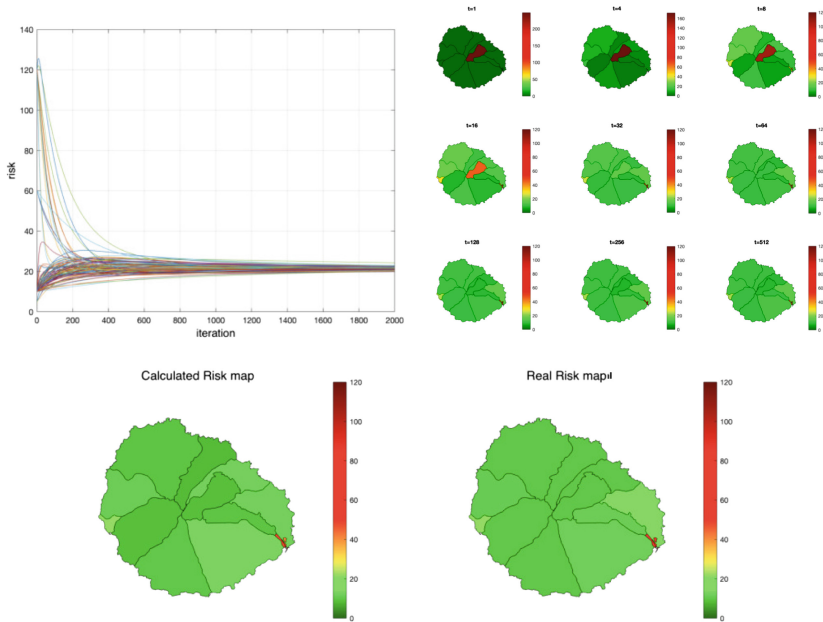


Fig. 4. Evolution of the consensus in the creation of a risk map. (Left) Convergence of the process. (Right) Evolution of the map calculated for one random node. (Bottom) Map created by consensus versus real risk map

Let us assume that people outside the risky areas do not move into them, and people who live in high-risk areas do not go out, depending on the risk map readings. The effect of having a risk map available and avoid areas with high risk does not reduce the total number of infections significantly. It reduces the peak but keeps the propagation active more days (see Fig. 5).

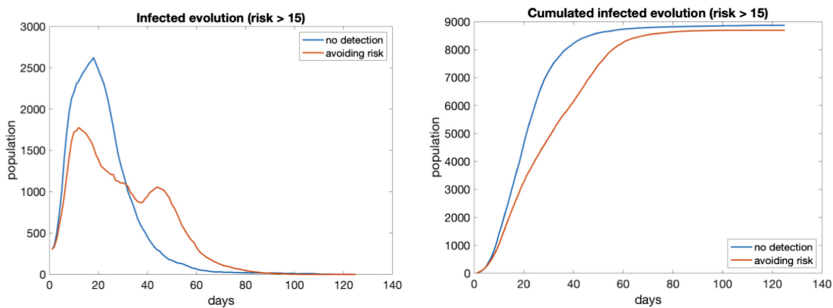


Fig. 5. Evolution of infected with and without considering the risk map. (Left) Evolution (Right) Cumulated

3.3 Evolution with Contact Tracing App Active

A problem with tracking applications is that they need a large percentage of the population using the app. Some works suggest that tracking applications need at least 60% of penetration to be effective [11]. We have simulated the propagation of COVID-16 in three scenarios: (1) no measures taken, (2) all infected detected and isolated, and (3) people with contact tracing app isolated in 48 h from the first symptom. To evaluate the impact of the penetration of the app, we have considered a 20%, 40%, 60%, and 80% of the total population using the app (Fig. 6).

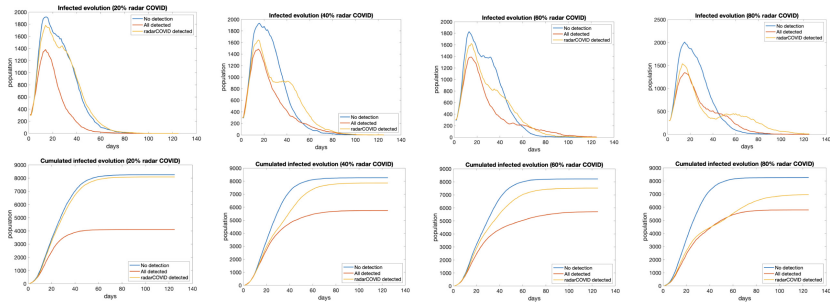


Fig. 6. Evolution of infected in three scenarios: no isolation (blue), total isolation (red) and isolation for traced users (yellow), from 20% to 80% of users. (top) total infected by day (bottom) cumulated infections

The effect of contact tracing apps in the propagation is almost irrelevant, with 20% of users. There are just little differences between 40% and 60%, so probably it is not necessary to arrive at this value. With 80%, the transmission is almost controlled.

3.4 Contact Tracing and Risk Maps Combined

Finally, we have tested the combination of risk maps and contact tracing. The behavior of the people would be

- if you live in an area with medium or high risk, you do not go out of it
- if you live in a low-risk area, do not go to risky ones
- tracing app notifies exposure in 48 h. If the person receives an alert, then is isolated

Five scenarios are analyzed: no isolation, total isolation, limited movement by risk map, isolation by contact tracing, and risk map plus contact tracing. Considering the case with 3,000 users (15% of the population of La Gomera), we see that contact tracing or risk maps have a low effect on their own in controlling the propagation (Fig. 7). However, with both strategies combined, we obtain a significant reduction in total infected, reducing a 50%.

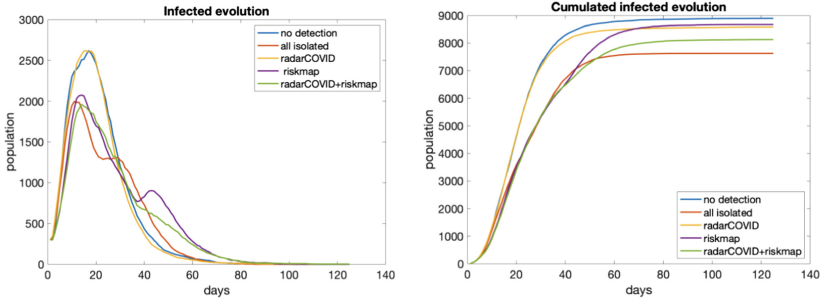


Fig. 7. Evolution if the number of infections with different technological solutions applied with 15% of penetration

4 Conclusions

Technology can be an essential ally to control the transmission of COVID-19. Nevertheless, concerns about privacy and the possible use of the data after the pandemic have made it difficult to implant technological solutions.

This work proposes an alternative for users to create risk maps collaboratively. This approach executes a consensus process that uses local information and data from the direct neighbors to calculate the value of a shared function. In our case, the values are the risk index of the different districts that form the town. Close contacts (family, colleagues, and friends) define the network relationships, whom we warned about being infected. The data exchanged is an aggregation, and it is not possible to reidentify the personal information. At the end of the process, all the participants obtain the same copy of the complete risk map. However, constraining the movements using the information on risk maps reduces the peak and smooth the evolution of the infection.

The use of contact tracing apps needs a considerable proportion of active users to work. Nevertheless, the combination of the information from risk maps to avoid areas with a high index of infections and alerts of exposure obtain good results even with relatively low penetration of the apps.

Acknowledgements. This research was supported by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215, and by the Spanish Ministry of Science, Innovation and Universities (MICIU) under Contract No. PGC2018-093854-B-I00b and RTI2018-095390-B-C32.

References

1. Apple, Google: Exposure notifications: using technology to help public health authorities fight covid'19, 01 June 2020. <https://www.google.com/covid19/exposurenotifications/>
2. Apple, Google: Privacy-preserving contact tracing, 01 June 2020. <https://www.apple.com/covid19/contacttracing>

3. Castelluccia, C., Bielova, N., Boutet, A., Cunche, M., Lauradoux, C., Le Métayer, D., Roca, V.: ROBERT: ROBust and privacy-presERving proximity Tracing, May 2020. <https://hal.inria.fr/hal-02611265/file/ROBERT-v1.1.pdf>
4. Chan, J., Foster, D., Gollakota, S., Horvitz, E., Jaeger, J., Kakade, S., Kohno, T., Langford, J., Larson, J., Sharma, P., Singanamalla, S., Sunshine, J., Tessaro, S.: Pact: privacy sensitive protocols and mechanisms for mobile contact tracing (2020). [arXiv:2004.03544](https://arxiv.org/abs/2004.03544) [cs.CR]
5. Dominguez-Garcia, A.D., Hadjicostis, C.N.: Distributed matrix scaling and application to average consensus in directed graphs. *IEEE Trans. Autom. Control* **58**(3), 667–681 (2013)
6. González, M., Hidalgo, C., Barabási, A.: Understanding individual human mobility patterns. *Nature* **453**, 779–782 (2008)
7. Olfati-Saber, R., Fax, J.A., Murray, R.M.: Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* **95**(1), 215–233 (2007)
8. O’Neill, P.H., Ryan-Mosley, T., Johnson, B.: A flood of coronavirus apps are tracking us. Now it’s time to keep track of them, 07 May 2020. <https://www.technologyreview.com/2020/05/07/1000961/launching-mittr-covid-tracing-tracker/>
9. Pan-European privacy-preserving proximity tracing, 01 June 2020. <https://www.pepp-pt.org/>
10. Rivest, R.L., Weitzner, D.J., Ivers, L.C., Soibelman, I., Zissman, M.A.: Pact: private automated contact tracing, 01 June 2020. pact.mit.edu
11. Hinch, R., Probert, W., Nurtay, A., Kendall, M., Wymant, C., Hall, M., Lythgoe, K., Cruz, A.B., Zhao, L., Stewart, A., Ferretti, L., Parker, M., Meroueh, A., Mathias, B., Stevenson, S., Montero, D., Warren, J., Mather, N.K., Finkelstein, A., Abeler-Dörner, L., Bonsall, D., Fraser, C.: Effective Configurations of a Digital Contact Tracing App: A report to NHSX. University of Oxford, 16 April 2020
12. Simko, L., Calo, R., Roesner, F., Kohno, T.: Covid-19 contact tracing and privacy: studying opinion and preferences (2020). [arXiv:2005.06056](https://arxiv.org/abs/2005.06056) [cs.CR]
13. The European Data Protection Board: Guidelines 04/2020 on the use of location data and contact tracing tools in the context of the COVID-19 outbreak, 01 June 2020. https://edpb.europa.eu/our-work-tools/our-documents/usmernenia/guidelines-042020-use-location-data-and-contact-tracing_en
14. Troncoso, C., Payer, M., Hubaux, J.P., Salath, M., Larus, J., Bugnion, E., Lueks, W., Stadler, T., Pyrgelis, A., Antonioli, D., Barman, L., Chatel, S., Paterson, K., apkun, S., Basin, D., Beutel, J., Jackson, D., Preneel, B., Smart, N., Singelee, D., Abidin, A., Guerses, S., Veale, M., Cremers, C., Binns, R., Cattuto, C.: Decentralized privacy-preserving proximity tracing, April 2020. <https://github.com/DP-3T/documents/blob/master/DP3T>
15. Vinuesa, R., Theodorou, A., Battaglini, M., Dignum, V.: A socio-technical framework for digital contact tracing. *Results Eng.* **8**, 100163 (2020)
16. WeTrace: Privacy-preserving bluetooth covid-19 contract tracing application, 01 June 2020. <https://wetrace.ch/>
17. Yang, Z., Zeng, Z., Wang, K., Wong, S.S., Liang, W., Zanin, M., Liu, P., Cao, X., Gao, Z., Mai, Z., Liang, J., Liu, X., Li, S., Li, Y., Ye, F., Guan, W., Yang, Y., Li, F., Luo, S., Xie, Y., Liu, B., Wang, Z., Zhang, S., Wang, Y., Zhong, N., He, J.: Modified SEIR and AI prediction of the epidemics trend of covid-19 in china under public health interventions. *J. Thorac. Dis.* **12**(3) (2020)

Dynamics on/of Networks



Distributed Algorithm for Link Removal in Directed Networks

Azwirman Gusrialdi^(✉)

Faculty of Engineering and Natural Sciences, Tampere University,
33014 Pirkanmaa, Finland
azwirman.gusrialdi@tuni.fi

Abstract. This paper considers the problem of removing a fraction of links from a strongly connected directed network such that the largest (in module) eigenvalue of the adjacency matrix corresponding to the network structure is minimized. Due to the complexity of the problem, an effective and scalable algorithm based on eigenvalue sensitivity analysis is proposed in the literature to compute the suboptimal solution to the problem. However, the algorithm requires knowledge of the global network structure and does not preserve strong connectivity of the resulting network. This paper proposes distributed algorithms which allow distributed implementation of the previously mentioned algorithm by relying solely on local information on the network topology while guaranteeing strong connectivity of the resulting network. A numerical example is provided to demonstrate the proposed distributed algorithm.

Keywords: Link removal · Strongly connected directed graph · Distributed algorithm · Optimization

1 Introduction

It is well-known that the dominant (largest in module) eigenvalue of the so-called adjacency matrix associated with a network plays an important role in the dissemination of an entity such as disease or information in both unidirectional and bidirectional networks. In other words, it determines whether the dissemination process will become an epidemic [4, 9, 14, 16, 19]. While there are several factors which affect dissemination process of an entity including the intrinsic property of the entity and the network topology, in this paper we assume that we could only modify the network structure where the entity spreads on. In particular, we focus on the problem of removing a fraction of links from a network in order to contain the dissemination by minimizing dominant eigenvalue of the network's adjacency matrix. The removal of links can be interpreted as controlling the interaction between people or cities in a country in order to slow the spread of disease when a vaccine is not yet available.

It is known that removing a fraction of links from a network to minimize the dominant eigenvalue of the adjacency matrix is a NP-hard problem [17].

In order to address this issue, several works have focused on developing strategies to approximate and compute sub-optimal solution to this problem for both unidirectional and bidirectional networks, see for example [1, 4, 12, 17]. An effective and scalable algorithm based on eigenvalue sensitivity analysis is presented in [4] to minimize dominant eigenvalue of the adjacency matrix by removing some links from a directed network. Specifically, an optimization problem involving the left and right eigenvectors corresponding the dominant eigenvalue is formulated to compute the sub-optimal solution. Note that the previously mentioned work assume that the global network structure is available and known to the designer. However, in practice the global network structure may not be available or may be very hard to obtain in a centralized manner due to geographical constraint or privacy concerns [10, 11]. In addition to the availability of information on global network structure, the previously mentioned work do not take into account the (strong) connectivity of the network after the link removal. In some cases, it is desirable to preserve the (strong) connectivity of a network, for example so that important information can still be passed to all the users/nodes in the network or goods can still be transported between cities. Note that in [8], distributed algorithms which do not require knowledge of global network structure are proposed to remove a fraction of links from a network while guaranteeing the connectivity of the resulting network. However, the result is only limited to the case of bidirectional or undirected network.

The contribution of this paper is the development of distributed algorithms to compute the sub-optimal solution to link removal problem in a directed network while preserving strong connectivity of the resulting network. Specifically, matrix perturbation approach proposed in the literature is combined with novel distributed algorithms to estimate both the left and right dominant eigenvectors of the adjacency matrix to decide the candidate link to be removed. Furthermore, distributed verification algorithm is proposed to check whether a strongly connected directed network remains to be strongly connected after removing a fraction of links. This paper also generalizes the results presented in [8]. The proposed distributed algorithms can also readily be applied to the link addition problem whose goal is to maximize dominant eigenvalue of the adjacency matrix.

The organization of this paper is as follows: preliminaries followed by the problem formulation are presented in Sect. 2. The proposed distributed algorithms for link removal in directed networks are described in Sect. 3. A numerical example to demonstrate the proposed distributed strategy is provided in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Problem Statement

We first provide a brief overview of graph theory and well-known results used to develop distributed link removal strategy followed by the problem formulation.

2.1 Notation and Preliminaries

Let \mathbb{R} be the set of real numbers and vector $\mathbf{1}_n \in \mathbb{R}^n$ denote the column vector of all ones. Furthermore, $\text{diag}(a) \in \mathbb{R}^{n \times n}$ represents the diagonal matrix with

the elements of vector $a \in \mathbb{R}^n$ on its diagonal. For a given set \mathcal{V} , $|\mathcal{V}|$ denotes the number of the elements in this set.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph (digraph) with a set of nodes $\mathcal{V} = \{1, 2, \dots, n\}$ and a set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. An edge $(j, i) \in \mathcal{E}$ denotes that node i can obtain information from node j . The set of in-neighbors of node i is denoted by $\mathcal{N}_{\mathcal{G},i}^{\text{in}} = \{j | (j, i) \in \mathcal{E}\}$. Similarly, the set of out-neighbors of node i is denoted by $\mathcal{N}_{\mathcal{G},i}^{\text{out}} = \{j | (i, j) \in \mathcal{E}\}$. The directed graph \mathcal{G} is *strongly connected* if every node can be reached from any other nodes by following a set of directed edges. For a matrix $C \in \mathbb{R}^{n \times n}$, let $[C]_{i*}$ and $[C]_{*i}$ represent vectors whose elements are equal to the i -th row and column of C respectively. Let us denote the dominant (i.e., largest in module) eigenvalue of matrix C as $\lambda(C)$. The adjacency matrix associated with digraph \mathcal{G} , denoted by $A(\mathcal{G}) \in \mathbb{R}^{n \times n}$ is defined as

$$[A(\mathcal{G})]_{ij} = \begin{cases} 1 & \text{if } i \neq j \text{ and } (j, i) \in \mathcal{E}, \\ 0, & \text{otherwise} \end{cases}$$

where $[A]_{ij}$ denote the element in the i -th row and j -th column of matrix A . The proposed algorithm can also be applied to adjacency matrix whose rows are defined using the out-neighbors of node i . Matrix $C \in \mathbb{R}^{n \times n}$ is nonnegative (i.e., $C \geq 0$) if all its elements are nonnegative. A nonnegative matrix C is irreducible if and only if $(I_n + C)^{n-1} > 0$ where $I_n = \text{diag}(\mathbf{1}_n)$. In addition, matrix C is primitive if it is irreducible and has at least one positive diagonal element.

Finally, we review a max-consensus algorithm. Consider a strongly connected digraph \mathcal{G} with n nodes and let us assign state $x_i(t) \in \mathbb{R}$ to each node of \mathcal{G} . If each node executes the following max-consensus algorithm [13]

$$x_i(t+1) = \max_{j \in \mathcal{N}_{\mathcal{G},i}^{\text{in}} \cup \{i\}} x_j(t) \quad (1)$$

then $x_i(t) = x_j(t) = \max_i x_i(0)$ for $i, j \in \mathcal{V}$, $\forall t \geq l$ where l is the maximum of the shortest path length between any pair of nodes in \mathcal{G} .

2.2 Problem Formulation

Consider an n node network whose connection is given by a (unweighted) strongly connected directed graph $\mathcal{G}_0 = \{\mathcal{V}, \mathcal{E}_0\}$. From Perron-Frobenius theorem, it can be observed that $\lambda(A(\mathcal{G}_0))$ is real, strictly positive and simple [2]. For the sake of simplicity, we assume that the nodes know the network's size n . Otherwise, its value can be estimated distributively using the methods proposed in the literature, see for example [15]. Our objective is to remove at most m_e number of links $\Delta\mathcal{E}^-$ from \mathcal{E}_0 such that dominant eigenvalue of the adjacency matrix of resulting graph $\mathcal{G}_{m_e} = \{\mathcal{V}, \mathcal{E}_0 \setminus \Delta\mathcal{E}^-\}$ is minimized while also guaranteeing that \mathcal{G}_{m_e} remains to be strongly connected. The problem can be formally formulated as the following optimization problem:

$$\begin{aligned} & \min_{\Delta\mathcal{E}^- \subseteq \mathcal{E}_0} \lambda(A(\mathcal{G}_{m_e})), \\ & \text{s.t.} \quad |\Delta\mathcal{E}^-| \leq m_e, \\ & \quad \mathcal{G}_{m_e} \text{ is strongly connected.} \end{aligned} \quad (\text{P1})$$

Solving optimization (P1) requires global knowledge on the network topology \mathcal{G}_0 . However, in practice the global network topology \mathcal{G}_0 is often unknown or not available due to geographical constraint or privacy reasons such as in social network. Motivated by this issue, we impose the following constraint for the remaining of the paper.

Constraint 1. *The overall network topology \mathcal{G}_0 is not available. In addition, node i can only receive information via a communication network from nodes in the set $\mathcal{N}_{\mathcal{G}_0,i}^{in}$ and knows the set $\mathcal{N}_{\mathcal{G}_0,i}^{out}$.*

The absence of information on the overall network topology makes it impossible to solve (P1) in a centralized manner. Furthermore, optimization (P1) is a combinatorial problem whose complexity increases exponentially with the network size. Therefore, we are interested in developing a distributed strategy to compute the suboptimal solution to (P1) as stated in the following problem.

Problem 1. Assume that graph \mathcal{G}_0 is strongly connected. Find a suboptimal solution or an upper bound to the solution to optimization (P1) under Constraint 1.

Remark 1. Flooding strategy where local information of each node is passed such that every node knows the overall network topology is not locally adaptable when the network topology changes. Moreover, if the information on the node's neighbors is private, this strategy will violate the individual node's privacy.

3 Main Result

In order to solve Problem 1, we first adopt the strategy based on matrix perturbation theory presented in [4, 8]. Using matrix perturbation theory, for a graph with a large spectral gap (i.e., difference between the largest and second largest eigenvalue in magnitude) we can write

$$\lambda(A(\mathcal{G}_{m_e})) = \lambda(A(\mathcal{G}_0)) - \frac{\nu_0^T \Delta A^- w_0}{\nu_0^T w_0} + O(\|\Delta A^-\|^2) \quad (2)$$

where ΔA^- denotes the adjacency matrix corresponding to the graph whose links are given by $\Delta \mathcal{E}^-$. Moreover, ν_0, w_0 are the dominant left and right eigenvectors corresponding to $\lambda(A(\mathcal{G}_0))$, respectively. Due to the large spectral gap, we can neglect the higher order term in (2) and thus minimizing $\lambda(A(\mathcal{G}_{m_e}))$ is equivalent to maximizing $\nu_0^T \Delta A^- w_0 / (\nu_0^T w_0)$. Defining the labeling $\ell_{\mathcal{G}_0} \in \{1, \dots, |\mathcal{E}_0|\}$ on the edges of graph \mathcal{G}_0 , matrix ΔA^- can be written as $\Delta A^- = \sum_{\ell_{\mathcal{G}_0}=1}^{|\mathcal{E}_0|} y_{\ell_{\mathcal{G}_0}} A_{\ell_{\mathcal{G}_0}}$ where $A_{\ell_{\mathcal{G}_0}}$ is a matrix with all zeros entries except for the ij th entry corresponding to the edge of label $\ell_{\mathcal{G}_0} \sim (j, i)$ which is equal to 1. Furthermore, $y_{\ell_{\mathcal{G}_0}} \in \{0, 1\}$ where $y_{\ell_{\mathcal{G}_0}} = 1$ means that the edge $\ell_{\mathcal{G}_0}$ in \mathcal{E}_0 is removed. Problem 1 can then

be formulated as the following optimization problem

$$\begin{aligned}
 & \max_{y \in \{0,1\}^{|\mathcal{E}_0|}} \frac{1}{\nu_0^T w_0} \sum_{\ell_{\mathcal{G}_0}=1}^{|\mathcal{E}_0|} y_{\ell_{\mathcal{G}_0}} \nu_{0,i} w_{0,j} \\
 & \text{s.t.} \quad \mathcal{G}_{m_e} \text{ is strongly connected,} \\
 & \quad \mathbf{1}_{|\mathcal{E}_0|}^T y \leq m_e,
 \end{aligned} \tag{P2}$$

where $\nu_{0,i}$ and $w_{0,i}$ respectively denotes the i -th element of left eigenvector ν_0 and w_0 associated with $\lambda(A(\mathcal{G}_0))$. In addition, the vector $y = [y_1, \dots, y_{|\mathcal{E}_0|}]^T$. The analysis of optimality gap between the solutions obtained by solving (P2) and (P1) is discussed in [4]. In order to solve (P2), Note that ν_0, w_0 cannot be directly computed and whether graph \mathcal{G}_{m_e} is strongly connected cannot be directly checked since the global network topology \mathcal{G}_0 is not available.

Next, we present distributed algorithms performed at each node, given that the nodes have local computational capability, to solve (P2) under constraint 1. To this end, we first define a primitive matrix Q_0 given by

$$Q_0 = I_n + A(\mathcal{G}_0). \tag{3}$$

Since matrix Q_0 is primitive, it is known that there exists a real dominant and simple eigenvalue of Q_0 , denoted by $\lambda(Q_0)$ satisfying $\lambda(Q_0) > |\mu|$ for all other eigenvalues μ of Q_0 [2]. Hence, we have the following relationship: $\lambda(Q_0) = 1 + \lambda(A(\mathcal{G}_0))$. It can also be observed that both matrices Q_0 and $A(\mathcal{G}_0)$ share the same set of left and right eigenvectors (i.e., ν_0, w_0) which are both positive, up to rescaling [2].

3.1 Distributed Estimation of Dominant Right Eigenvector w_0

In this subsection we utilize power iteration method to estimate w_0 in a distributed manner. Specifically, each node performs the following iterations [6]:

$$\hat{w}_{0,i}(t+1) = \frac{1}{\|Q_0 \hat{w}_0(t)\|_\infty} \sum_{j \in \{\mathcal{N}_{\mathcal{G}_0,i}^{\text{in}} \cup i\}} [Q_0]_{ij} \hat{w}_{0,j}(t) \tag{4}$$

where $\hat{w}_{0,i}(t)$ denotes the local estimation of $w_{0,i}$ at the t -th iteration. Note that since $w_0 > 0$, each node can choose any initial condition $\hat{w}_{0,i} > 0$. Furthermore, since the graph is strongly connected, it is guaranteed that under update law (4) local estimate $\hat{w}_{0,i}(t)$ will asymptotically converge to $w_{0,i}$ for all nodes i . Note that by using max-consensus algorithm (1) and by setting $x_i(0) = \sum_{j \in \{\mathcal{N}_{\mathcal{G}_0,i}^{\text{in}} \cup i\}} [Q_0]_{ij} \hat{w}_{0,j}(t)$, each node will be able to compute $\|Q_0 \hat{w}_0(t)\|_\infty$ in a distributed manner. Therefore, update law (4) can then be implemented distributively by each node in the network. The nodes can implement the stopping criteria $\|\hat{w}_0(t) - \hat{w}_0(t-1)\|_\infty < \epsilon$ for a sufficiently small pre-defined threshold ϵ (to guarantee the estimation accuracy) which can also be checked in a distributed manner using max-consensus algorithm.

Remark 2. The normalization in (4) is performed to prevent the nonzero components in the iteration from becoming extremely large when $|\lambda| > 1$ or approaching zero if $|\lambda| < 1$. Hence, the normalization can be performed intermittently (which can be agreed by the nodes in advance before implementing the algorithm) since it has no effects on the convergence of power iteration method [5].

3.2 Distributed Estimation of Dominant Left Eigenvector ν_0

After estimating distributively the dominant right eigenvector w_0 , the next step is to estimate the dominant left eigenvector ν_0 in a distributed manner. In contrast to the dominant right eigenvector, distributed estimation of the dominant left eigenvector has received less attention in the literature. To this end, we depart from the following relationship

$$Q_0^T \nu_0 = \lambda(Q_0) \nu_0, \tag{5}$$

where Q_0^T denotes the transpose of matrix Q_0 . Each node can then distributively estimate ν_0 by solving (5) in a distributed fashion. First, observe that after estimating $w_{0,i}$ and from $Q_0 w_0 = \lambda(Q_0) w_0$, node i can estimate $\lambda(Q_0)$ according to

$$\lambda(Q_0) = \frac{[Q_0]_{i*}^T \hat{w}_0}{\hat{w}_{0,i}}. \tag{6}$$

Next, since node i knows $\mathcal{N}_{\mathcal{G}_0,i}^{out}$ it can construct the vector $[Q_0]_{*i}$ or $[Q_0^T]_{i*}$. In addition, after estimating $\lambda(Q_0)$ from (6), each node then estimates ν_0 by solving distributively a set of linear equations (5) which can be rewritten as

$$\underbrace{(Q_0^T - \lambda(Q_0)I_n)}_{\bar{Q}_0} \nu_0 = 0. \tag{7}$$

Specifically, the nodes cooperatively estimate ν_0 by performing the following iterations [18]:

$$\hat{\nu}_0^i(t+1) = \hat{\nu}_0^i(t) - P_i \left(\hat{\nu}_0^i(t) - \frac{1}{|\mathcal{N}_{\mathcal{G}_0,i}^{in}|} \sum_{j \in \mathcal{N}_{\mathcal{G}_0,i}^{in}} \hat{\nu}_0^j(t) \right) \tag{8}$$

where $\hat{\nu}_0^i(t)$ denotes the local estimation of ν_0 at node i at the t -th iteration and matrix P_i is defined as

$$P_i = I_n - [\bar{Q}_0]_{i*} ([\bar{Q}_0]_{i*}^T [\bar{Q}_0]_{i*})^{-1} [\bar{Q}_0]_{i*}^T$$

which depends on local information of node i and \bar{Q}_0 is defined in (7). It should be noted that in general the set of linear equations (7) has many solutions. In order for local estimation $\hat{\nu}_0^i$ for $i = \{1, \dots, n\}$ to converge to the same solution to (7), the initial condition of each node $\hat{\nu}_0^i(0)$ is chosen to minimize

$$\frac{1}{2} |\hat{\nu}_0^i(0) - b|^2 \text{ s.t. } [\bar{Q}_0]_{i*}^T \hat{\nu}_0^i(0) = 0 \tag{9}$$

for arbitrary vector $b > 0$ with $|\cdot|$ denotes the Euclidean norm. It is shown in [18] that under update law (8) whose initial conditions are chosen to minimize (9), all the nodes estimation $\hat{\nu}_0^i$ converge *exponentially fast* to the solution to (7) which is also the solution to: $\min_{\bar{Q}_0 \nu_0 = 0} \frac{1}{2} |\nu_0 - b|^2$. The settling time of update law (8) can be calculated similar to the calculation in [7]. Note that update law (8) utilizes the same communication network \mathcal{G}_0 as the one utilized to distributively estimate w_0 . Furthermore, in contrast to the estimation of w_0 presented in the previous subsection, node i will obtain the estimation of the full vector ν_0 instead of the i -th element $\nu_{0,i}$.

Remark 3. In comparison to distributed algorithm for estimating left and right eigenvectors corresponding to any irreducible matrices presented in [7] which requires each node to use memory $O(n^2)$ and to send n^2 values to its neighbors, the proposed distributed algorithm only requires to use memory $O(n)$ and to send n values to its neighbors. In addition, applying distributed estimation algorithms in [7] will reveal the global network structure to all nodes which may violate the privacy of each node. In contrast to [7], the proposed distributed algorithm respects the privacy in terms of the global network topology.

3.3 Distributed Verification of Digraph's Strong Connectivity

Let us assume that we remove a link $(j^*, i^*) \in \mathcal{E}_0$ from a strongly connected digraph \mathcal{G}_0 . In this subsection we present a distributed algorithm based on max-consensus protocol to verify whether the resulting network $\mathcal{G}_1 = \{\mathcal{V}, \mathcal{E}_0 \setminus (j^*, i^*)\}$ remains to be strongly connected. To this end, each node is assigned a new variable $x_i(t) \in \mathbb{R}$ whose initial value is first set to $x_i(0) = 0, i = \{1, \dots, n\}$. Given a candidate link to be removed (j^*, i^*) , node j^* then modifies its initial value into $x_{j^*}(0) = 1$ while the remaining nodes do not change their initial values. All the nodes then execute max-consensus protocol (1) on the graph $\mathcal{G}_1 = \{\mathcal{V}, \mathcal{E}_0 \setminus (j^*, i^*)\}$, that is node j^* does not send its information to node i^* when executing the update law (1). We then have the following result on the relation between the final values of $x_i(t)$ and the strong connectivity of graph \mathcal{G}_1 .

Lemma 1. *Given a strongly connected digraph \mathcal{G}_0 and a link $(j^*, i^*) \in \mathcal{E}_0$. Each node executes max-consensus protocol (1) on the graph $\mathcal{G}_1 = \{\mathcal{V}, \mathcal{E}_0 \setminus (j^*, i^*)\}$ with initial values $x_{j^*}(0) = 1$ and $x_m(0) = 0$ for all $m \neq j^*$. The graph $\mathcal{G}_1 = \{\mathcal{V}, \mathcal{E}_0 \setminus (j^*, i^*)\}$ is strongly connected if and only if $x_i(n) = 1$ for all $i \in \mathcal{V}$.*

Proof. For showing the necessity (\implies), since the graph $\mathcal{G}_1 = \{\mathcal{V}, \mathcal{E}_0 \setminus (j^*, i^*)\}$ is strongly connected, it is shown in [13] that under max-consensus protocol (1) all nodes will converge to $\max_i x_i(0)$ which is equal to 1. Next, we show the sufficiency (\impliedby). To do this note that the graph \mathcal{G}_0 is strongly connected. The removal of link (j^*, i^*) thus may result in that there exists no direct or indirect path from node j^* to node i^* . However, since we have $x_i(n) = 1$ under update law (1) for all nodes i in the network, this means that there is at least an indirect path from nodes j^* to i^* . Hence, the resulting graph $\mathcal{G}_1 = \{\mathcal{V}, \mathcal{E}_0 \setminus (j^*, i^*)\}$ remains to be strongly connected which completes the proof.

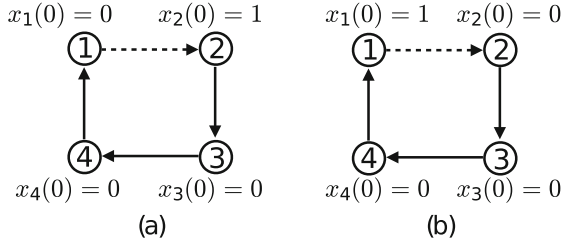


Fig. 1. Each node executes max-consensus protocol (1) on the graph $\mathcal{G}_1 = \{\mathcal{V}, \mathcal{E}_0 \setminus (1, 2)\}$. (a) the graph \mathcal{G}_1 is not strongly connected even though $x_i(n) = 1$ for all nodes i when $x_2(0) = 1$ and $x_1(0) = 0$; (b) graph \mathcal{G}_1 is not strongly connected since $x_i(n) = 0$ for all nodes $i \neq 1$ when $x_1(0) = 1$ and $x_2(0) = 0$

Remark 4. For the result in Lemma 1 to hold it requires the graph \mathcal{G}_0 to be strongly connected. In other words, the resulting graph $\mathcal{G}_1 = \{\mathcal{V}, \mathcal{E}_0 \setminus (j^*, i^*)\}$ may not be necessarily strongly connected even though $x_i(n) = 1$ for all $i \in \mathcal{V}$ if the graph \mathcal{G}_0 is not strongly connected.

Remark 5. In contrast to the case of undirected network presented in [8], in the case of directed network the initial values in (1) cannot be chosen randomly between nodes i^* and j^* in order to check whether the resulting network is still strongly connected as illustrated in Fig. 1.

After each node executes update law (1) for n iterations with initial values described in Lemma 1, node i^* then checks whether $x_{i^*}(n) = 1$. If $x_{i^*}(n) = 1$, it needs to notify node j^* that the network remains to be strongly connected in the removal of link (j^*, i^*) . To this end, each node is assigned with additional scalar variable $f_i(t)$. If the graph \mathcal{G}_1 is strongly connected (resp. not strongly connected), the initial values of f_i are set to $f_{i^*}(0) = 1$ (resp. $f_{i^*}(0) = -1$) and $f_m(0) = 0$ for all $m \neq i^*$. The nodes then again execute (1) on graph \mathcal{G}_0 with the previously described $f_i(0)$ and after n iterations, node j^* checks if $x_{j^*}(n) = 1$ (resp. $x_{j^*}(n) = 0$) then it will know that the graph \mathcal{G}_1 remains to be strongly connected (resp. will not be strongly connected) after the removal of the link (j^*, i^*) .

The strong connectivity of the resulting graph after removal of multiple links can then be distributively checked by iteratively applying the result in Lemma 1.

3.4 The Complete Distributed Link Removal Algorithm

After describing key elements required to develop the distributed algorithm, pseudo-code of the complete distributed link removal algorithm to solve optimization problem (P2) is summarized in Algorithm 1.

Note that in steps 6 and 7 of Algorithm 1 it is assumed that the final estimation of left and right eigenvectors ν_0, w_0 have been normalized. For the left eigenvector ν_0 , since each node has the estimation of the vector ν_0 , i.e., $\hat{\nu}_0^i$, it

Algorithm 1. Distributed algorithm to solve optimization problem (P2)

Require: \mathcal{G}_0 is strongly connected, node j can receive information from $\mathcal{N}_{\mathcal{G}_0,j}^{\text{in}}$ and knows $\mathcal{N}_{\mathcal{G}_0,j}^{\text{out}}, n, m_e$

- 1: $y = [0, \dots, 0]^T \in \mathbb{R}^{|\mathcal{E}_0|}$
- 2: node j estimate $w_{0,j}$ using (4) whose estimation is given by $\hat{w}_{0,j}$
- 3: node j estimate the vector ν_0 using (8) whose estimation is given by $\hat{\nu}_0^j$
- 4: initialize $p = 0$
- 5: **while** $p \leq m_e - 1$ **do**
- 6: node j independently computes $(j, i^c) = \operatorname{argmax} \hat{\nu}_{0,i}^j \hat{w}_{0,j}$ for $i \in \mathcal{N}_{\mathcal{G}_p,j}^{\text{out}}$
- 7: all nodes compute $(j^*, i^*) = \operatorname{argmax} \hat{\nu}_{0,i^c}^j \hat{w}_{0,j}$ with (j, i^c) obtained in the previous step using max-consensus (1) with $x_j(0) = \hat{\nu}_{0,i^c}^j \hat{w}_{0,j}$
- 8: check strong connectivity of $\mathcal{G}_{p+1} = (\mathcal{V}, \mathcal{E}_p \setminus (j^*, i^*))$ using distributed algorithm described in Subsection 3.3
- 9: **if** \mathcal{G}_{p+1} is not strongly connected **then**
- 10: back to steps 6–8 where node j^* excludes the link (j^*, i^*) when solving the optimization problem in step 6
- 11: **if** $\mathcal{N}_{\mathcal{G}_p,i^*}^{\text{out}} = \emptyset$ for all i^* **then**
- 12: **break**
- 13: **end if**
- 14: **else**
- 15: continue to step 17
- 16: **end if**
- 17: $p \leftarrow p + 1$
- 18: update $\mathcal{G}_p = \{\mathcal{V}, \mathcal{E}_{p-1} \setminus (j^*, i^*)\}$
- 19: $y_{\ell_{\mathcal{G}_0}^*} = 1$ where $\ell_{\mathcal{G}_0}^* \sim (j^*, i^*)$
- 20: **end while**

can then normalize the estimation $\hat{\nu}_0^i$ independently. On the other hand, since node i only has the estimation of the i -th element of right eigenvector w_0 , it then needs to normalize \hat{w}_0 cooperatively with the rest of the nodes. To this end, the norm $\|\hat{w}_0\|$ defined as $\|\hat{w}_0\| = \sqrt{\hat{w}_{0,1}^2 + \dots + \hat{w}_{0,n}^2}$ can also be written as

$$\|\hat{w}_0\| = \sqrt{n \left(\frac{\hat{w}_{0,1}^2 + \dots + \hat{w}_{0,n}^2}{n} \right)} = \sqrt{n \hat{w}_0^{\text{ave}}}.$$

If the nodes can compute \hat{w}_0^{ave} distributively and given that they know n , each node can then compute $\|\hat{w}_0\|$. Specifically, the nodes can compute \hat{w}_0^{ave} in a distributed manner using the finite-time average consensus algorithm proposed in the literature, e.g., [3] by setting its initial value as $\hat{w}_{0,i}^2$.

4 An Illustrative Example

In this section, we demonstrate the proposed distributed algorithm to compute solution to Problem 1. Consider a strongly connected digraph \mathcal{G}_0 consisting

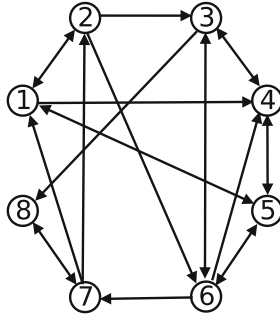


Fig. 2. A strongly connected directed graph used in the simulation

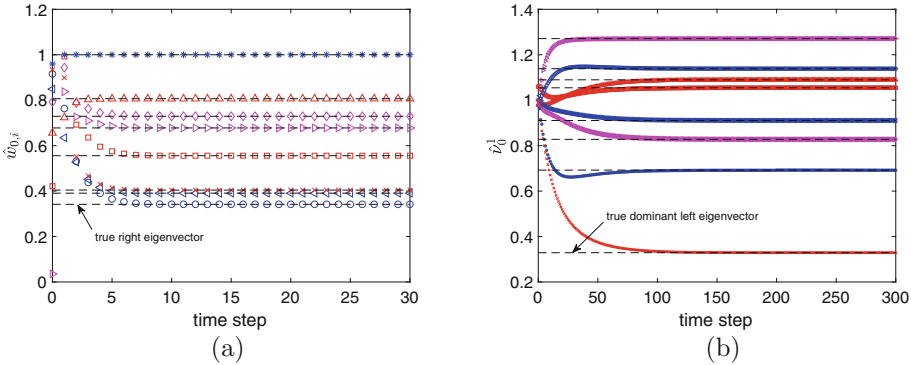


Fig. 3. (a) Estimated right eigenvector $\hat{w}_{0,i}$ corresponding to $\lambda(Q_0)$ with $Q_0 = A(\mathcal{G}_0) + I_n$ by each node (denoted by the markers) using power iteration method in (4). The estimation converge in less than 20 time steps; (b) Estimation of left eigenvector ν_0 corresponding to $\lambda(Q_0)$ by node 1 (i.e., $\hat{\nu}_0^1$)

of eight nodes shown in Fig. 2 where each node may represent for example a city/state or a person. The number of links to be removed m_e is varied between 1 and 4. We choose a small size network so that the comparison with the centralized brute-force search approach, which in general is NP-hard, becomes possible. Interested reader is referred to the simulation results in [4] for the performance evaluation of solution to optimization problem (P2), without connectivity constraint, on real large graphs.

We apply Algorithm 1 to find solution to optimization problem (P2) for each m_e . As illustrated in Fig. 3a, the estimation $\hat{w}_{0,i}$ converges in less than 20 time steps to the true (unnormalized) right eigenvector $w_{0,i}$. Next, each node distributively estimates the left eigenvector ν_0 using update law (8). Figure 3b depicts the left eigenvector estimation of $A(\mathcal{G}_0)$ by node 1. As can be observed, the local estimation by each node converges to the true (unnormalized) left eigenvector ν_0 . After estimating the eigenvectors (and normalizing them), the nodes then compute the candidate edge to be removed and check the strong connectiv-

Table 1. Comparison of solution using different strategies

m_e	Algorithm 1 Optimization (P2)		Iterative link removal Optimization (P2)		Brute-force search Optimization (P1)	
	$\Delta\mathcal{E}^-$	$\lambda(A(\mathcal{G}_{m_e}))$	$\Delta\mathcal{E}^-$	$\lambda(A(\mathcal{G}_{m_e}))$	$\Delta\mathcal{E}^-$	$\lambda(A(\mathcal{G}_{m_e}))$
1	(4,5)	2.5209	(4,5)	2.5209	(4,5)	2.5209
2	(4,5), (5,6)	2.3717	(4,5), (3,6)	2.2426	(4,5), (3,6)	2.2426
3	(4,5), (5,6) (3,6)	2.0826	(4,5), (3,6) (1,2)	2.0295	(1,2), (3,6) (5,6)	2.0135
4	(4,5), (5,6) (3,6), (5,1)	1.9728	(4,5), (3,6) (1,2), (5,6)	1.8216	(1,5), (3,6) (4,5), (7,2)	1.8111

ity of resulting graph. For comparison, we modify Algorithm 1 to iteratively and distributively remove one link at a time, that is after removing a link from the network we re-estimate the dominant left and right eigenvectors corresponding to the resulting network and compute the next link to be removed based on the new estimated dominant eigenvectors. In addition, to evaluate the optimality gap between the suboptimal and the global optimal solutions we also solve the original optimization problem (P1) by performing a brute-force search for each m_e and assuming that the global network topology is available.

The results are summarized in Table 1. First, it can be observed that for $m_e = 1$ the solutions to (P1) and (P2) are the same, i.e., the optimality gap is equal to zero. For the case of $m_e = 2, 3, 4$ there is a gap between the values of $\lambda(A(\mathcal{G}_{m_e}))$ corresponding to the solution obtained from Algorithm 1 and by applying brute-force search. However, the gap could be made smaller if we iteratively remove one link at a time for each m_e . In fact, when $m_e = 2$ the optimality gap between iterative link removal and brute force search is equal to zero, i.e., there is no performance loss in spite of the absence of the global network topology. Intuitively, one of the reasons is because by removing iteratively one link at a time, the matrix perturbation ΔA^- in (2) becomes sufficiently small so that the term $\nu_p^T \Delta A^- w_p / (\nu_p^T w_p)$ at iteration p could accurately predict the movement of eigenvalue $\lambda(A(\mathcal{G}_p))$ when it is perturbed by ΔA^- .

5 Conclusion

This paper proposes eigenvalue sensitivity based distributed algorithm to remove a fraction of links from a strongly connected directed network such that dominant eigenvalue of the adjacency matrix is minimized. In addition, the algorithm also guarantees that the resulting network remains to be strongly connected after the link removals. The proposed distributed algorithms consist of distributed estimation of both left and right eigenvectors corresponding to the largest (in module) eigenvalue of the adjacency matrix together with distributed verification algorithm to check whether a network is strongly connected after removal of a

link. A numerical example demonstrates the implementation and efficacy of the proposed distributed algorithm.

Acknowledgement. This work is supported by Academy of Finland under academy project decision number 330073.

References

1. Bishop, A.N., Shames, I.: Link operations for slowing the spread of disease in complex networks. *Europhys. Lett.* **95**(18005) (2011)
2. Bullo, F.: *Lectures on Network Systems*. CreateSpace, 1 edn. (2018). <http://motion.me.ucsb.edu/book-Ins>. with contributions by J. Cortes, F. Dorfler, and S. Martinez
3. Charalambous, T., Yuan, Y., Yang, T., Pan, W., Hadjicostis, C.N., Johansson, M.: Distributed finite-time average consensus in digraphs in the presence of time delays. *IEEE Trans. Control Netw. Syst.* **2**(4), 370–381 (2015)
4. Chen, C., Tong, H., Prakash, B.A., Eliassi-Rad, T., Faloutsos, M., Faloutsos, C.: Eigen-optimization on large graphs by edge manipulation. *ACM Trans. Knowl. Discovery Data (TKDD)* **10**(4), 49 (2016)
5. Ghaboussi, J., Wu, X.S.: *Numerical Methods in Computational Mechanics*. CRC Press (2016)
6. Golub, G.H., Van Loan, C.F.: *Matrix Computations*, 3rd edn. Johns Hopkins University Press, Baltimore (1996)
7. Gusrialdi, A., Qu, Z.: Distributed estimation of all the eigenvalues and eigenvectors of matrices associated with strongly connected digraphs. *IEEE Control Syst. Lett.* **1**(2), 328–333 (2017)
8. Gusrialdi, A., Qu, Z., Hirche, S.: Distributed link removal using local estimation of network topology. *IEEE Trans. Netw. Sci. Eng.* **6**(3), 280–292 (2019)
9. Li, C., Wang, H., Van Mieghem, P.: Epidemic threshold in directed networks. *Phys. Rev. E* **88**(6), 062802 (2013)
10. Li, N., Zhang, N., Das, S.: Preserving relation privacy in online social network data. *IEEE Internet Comput.* **15**(3), 35–42 (2011)
11. McDaniel, P., McLaughlin, S.: Security and privacy challenges in the smart grid. *Secur. Privacy IEEE* **7**(3), 75–77 (2009)
12. Milanese, A., Sun, J., Nishikawa, T.: Approximating spectral impact of structural perturbations in large networks. *Phys. Rev. E* **81**(4), 046112 (2010)
13. Nejad, B., Attia, S., Raisch, J.: Max-consensus in a max-plus algebraic setting: the case of fixed communication topologies. In: *International Symposium on Information, Communication and Automation Technologies*, pp. 1–7 (2009)
14. Prakash, B.A., Chakrabarti, D., Valler, N., Faloutsos, M., Faloutsos, C.: Threshold conditions for arbitrary cascade models on arbitrary networks. *Knowl. Inf. Syst.* **33**(3), 549–575 (2012)
15. Shames, I., Charalambous, T., Hadjicostis, C.N., Johansson, M.: Distributed network size estimation and average degree estimation and control in networks isomorphic to directed graphs. In: *Proceedings of Annual Allerton Conference on Communication, Control, and Computing*, pp. 1885–1892 (2012)
16. Van Mieghem, P., Van de Bovenkamp, R.: Non-markovian infection spread dramatically alters the susceptible-infected-susceptible epidemic threshold in networks. *Phys. Rev. Lett.* **110**(10), 108701 (2013)

17. Van Mieghem, P., Stevanović, D., Kuipers, F., Li, C., van de Bovenkamp, R., Liu, D., Wang, H.: Decreasing the spectral radius of a graph by link removals. *Phys. Rev. E* **84**, 016101 (2011)
18. Wang, X., Mou, S., Sun, D.: Improvement of a distributed algorithm for solving linear equations. *IEEE Trans. Ind. Electron.* **64**(4), 3113–3117 (2016)
19. Wang, Y., Chakrabarti, D., Wang, C., Faloutsos, C.: Epidemic spreading in real networks: an eigenvalue viewpoint. In: *Proceedings of 22nd International Symposium on Reliable Distributed Systems*, pp. 25–34 (2003)



Data Compression to Choose a Proper Dynamic Network Representation

Remy Cazabet^(✉)

Univ de Lyon, CNRS, Université Lyon 1, LIRIS, UMR5205, Villeurbanne, France
remy.cazabet@gmail.com

Abstract. Dynamic network data are now available in a wide range of contexts and domains. Several representation formalisms exist to represent dynamic networks, but there is no well known method to choose one representation over another for a given dataset. In this article, we propose a method based on data compression to choose between three of the most important representations: snapshots, link streams and interval graphs. We apply the method on synthetic and real datasets to show the relevance of the method and its possible applications, such as choosing an appropriate representation when confronted to a new dataset, and storing dynamic networks in an efficient manner.

Keywords: Temporal networks · Dynamic networks · Link streams · Information theory

1 Introduction

The analysis of dynamic networks is an important topic of research in the network science community. With the ubiquity of digital data collection, more and more relational data becomes available with a temporal component. However, the way to handle and model such data is still a research question. As discussed in several articles in the state of the art [5, 8, 10, 11], there are multiple ways to model the same original observations.

In this article, we propose a method to choose an appropriate dynamic graph model, among the three following ones: Sequence of Snapshots, Interval Graphs and Link Streams. The method we propose is based on the principle of maximizing data compression, i.e., minimizing the network description's encoding length.

In Sect. 2, we explain the rationale of our approach, and possible applications. In Sect. 3, we introduce the computation of the encoding cost of a dynamic network, for each representation. Section 4 present experiments on synthetic and real datasets. Finally, we conclude in Sect. 5.

2 Context and Motivation

Dynamic networks can be used to represent a variety of real-world phenomena, with widely different properties. For instance, some networks represent

interactions (e.g., e-mails, instant messages on Social Media, physical proximity, co-authoring in scientific networks, etc.), while some others represent persistent *relations* (e.g., follower/followee relations on tweeter, friendship relations, active collaborations in scientific networks, etc.). It is clear that those two types of networks are of very different nature, and should be modeled and analyzed in different ways. But the difference between those two types is not as obvious as it might seem at first sight: some interactions have a duration (e.g., phone calls, physical proximity, etc.), while the nature of some collected data might be ambiguous (sentimental relations between teens, collaboration on a scientific project, etc.).

Another source of difficulty is that data are often collected at a given temporal granularity, either for convenience or for technical constraints. For instance, scientific publications are often characterized by their publication year only, interaction logs might be rounded-up to the hour or even the day for privacy reasons, and large datasets such as friendship in Facebook are often collected at a low frequency, e.g., once a month or year.

For all these reasons, the choice of a model is often not as simple as knowing the nature of the studied data, but requires to look at the data properties.

2.1 The Different Models of Dynamic Networks

In this article, we will consider three types of dynamic network models, often used in the literature.

- Snapshot (SN): The network is represented as a sequence of graphs. Each graph corresponds to a point in time or is made of the sum of all interactions over a period.
- Link Stream (LS): The network is a collection of edges, each identified by a pair of node and a point in time
- Interval Graphs (IG): The network is a collection of edges, and each edge exists over a given time interval, identified by its start and end times.

While these representations might appear unrelated at first sight, they are in fact able to represent the same original data as long as time is provided as a *discrete* value, which is the case in most practical situations. For instance, if time is represented as a POSIX time (timestamp), it can be considered discrete, since there is a countable number of possible values between any two POSIX time.

The best way to understand how the same data can be represented by these different models is to take a practical example: the SocioPatterns projet [1] has collected several physical interaction datasets between individuals in various contexts, such as schools or hospitals. Wearable sensors collect every face-to-face interaction at a high frequency between a group of subjects over an extended period of time, from a few hours to a few days. For practical reasons, data collection is made for the whole setting every 20 s, in a synchronous fashion. The publicly provided data therefore consists in a file containing all those captured

interactions, as triplets $\langle T, U1, U2 \rangle$, with T the timestamp of the interaction, $U1$ and $U2$ the face-to-face individuals. There are therefore several ways to interpret such data:

- SN: Each 20s, a snapshot of all on-going conversations is captured. Each of these snapshots could be studied as a conventional static graph.
- LS: Each triplet is a link of the link stream, it corresponds to an observed interaction between individuals.
- IG: Individuals do not interact punctually but over periods of time, usually longer than 20 s. Technical reasons force to capture the state of these persistent interactions periodically every 20 s. To model more accurately the observed phenomenon, one should create continuous intervals for each pair of node interacting repeatedly every 20 s over a period, lasting from the first to the last observation of the series.

Any of these interpretations is valid a priori, so the choice of using one instead of another is usually based on practical reasons, e.g., to apply a method that requires to have the data in one format or another. For instance, dynamic community detection algorithms assume a specific network format: Snapshots in most cases (e.g., [13]), but also sometimes Link streams (e.g., [12, 17]) or Interval Graphs (e.g., [3, 6]).

2.2 Using Encoding Cost as a Selection Criterion

The principle that the best description of data is the description that minimizes the cost of its representation can be found in several areas of science, from Occam’s Razor to the Minimum Description Length [9] (MDL) principle.

For static networks, this principle could for instance be used to choose between a matrix representation, an edge-list representation and an adjacency list representation. For an unweighted, undirected network composed of n nodes and m edges, the cost (in bits) of a matrix representation is n^2 (a matrix of boolean values), while its corresponding representation as an edge list is $2m \log_2(n)$ –encoding each edge requires to encode each of its 2 nodes. It means that if the graph is sparse, $m \ll n^2$, the edge list representation is the most efficient, and vice versa. The adjacency list representation is beyond the scope of this article, but relatively similar to the adjacency list. A first implication is that we can choose the most appropriate way to store the graph in memory given its properties n and m , but, beyond this, it also provides hints on what can be done or not on this graph. For instance, in the community detection problem, methods using matrix factorization are little penalized by the density of the matrix, while an algorithm such as Louvain, designed for sparse graphs, only requires the neighborhoods of nodes, available in an adjacency list representation. Therefore, the best way to encode a graph also gives us hints on how to handle it.

Note that in this paper, we limit ourselves to the comparison of encoding scheme that depends only on the number of nodes, edges and temporal information, and not on other properties such as the degree distribution, that could

also be optimized with techniques such as Huffman Coding. When dealing with dynamic graphs, we will also make the assumption in our representation that cumulated graphs of networks to represent are sparse, since this is the most common setting. We therefore propose representations which are extensions of edge lists rather than adjacency matrices.

2.3 Applications

The method introduced in this article is implemented in *tnetwork*¹, a python library to manipulate temporal networks. The first application is to automatically choose the most efficient in-memory representation for a temporal network loaded from a file containing triplets $\langle T, U1, U2 \rangle$, as is the case for temporal networks shared by the SocioPatterns project and those available on the *Network Repository*² website [14].

The method is also used in the library to choose the most parsimonious representation when saving temporal networks created using the library, such as random temporal networks with community structure.

Beyond these memory-related applications, knowing the most appropriate representation also tells the practitioner how to efficiently manipulate their data. For instance, if the snapshot representation is inefficient to represent a dynamic network, it is unwise to analyze each period as an independent snapshot, for instance computing centralities or detecting communities for each of them. Reciprocally, if the network is poorly represented as a link stream, it is unwise to apply methods expecting such a graph, e.g. the community detection method introduced in [12].

3 Temporal Network Encoding Cost

Following [11], let's define our dynamic network as a link stream $L = (T, V, E)$, where T is the list of possible times, V the list of possible vertices, and E triplets composed of two vertices and a time. We also define the aggregated graph $G^a(L) = (V, E^a)$, such as E^a is the list of pairs of nodes (edges) that appear at least once in the link stream L . The encoding cost of a given temporal network with a given representation depends mainly on three properties:

- $m = |E^a|$ is the number of different edges to appear at least once
- $e = |E|$ is the total number of events, i.e., links in the link stream
- $t = |T|$ is the total number of different times during which at least one event occurs in the dynamic network.

We also need two partial encoding costs:

- $I^t = \log_2(t)$ is the cost to encode one time information
- $I^m = 2 \log_2(|V|)$ is the cost to encode one node pair

¹ <https://tnetwork.readthedocs.io>.

² <http://networkrepository.com>.

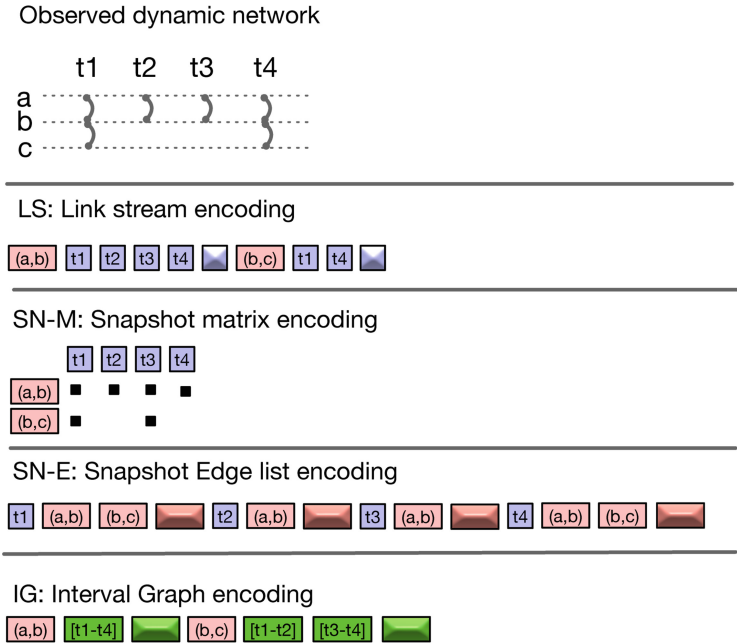


Fig. 1. Illustration of the chosen encoding strategies. From an observed dynamic network, each strategy encodes node pairs (red) and temporal information (blue/green). Textured blocs correspond to STOP symbols to mark the end of a series of unknown length.

The cost of encoding nodes themselves is a constant for a given network and is thus ignored in the rest of this paper.

Link Stream Encoding. For this encoding, we list, for each pair of nodes, the list of timestamps it appears in. The total encoding cost of a graph is:

$$I^{ls} = mI^m + eI^t + mI^t$$

where the first element is the cost of encoding the edges, the second element encodes the timestamps, and the last encode stop sections to signal the end of a list of times.

We formalize below the cost of encoding a dynamic network using four representations. A summary of these representations can be found in Fig. 1.

SN Encoding. We consider two ways of encoding snapshot sequences, the first one to represent snapshots that have most of their edges in common, and the other one for snapshots that are few of their edges in common.

In this first snapshot representation, we encode data as a matrix, whose lines correspond to pairs of nodes and columns to timestamps. A unique bit is required to indicate if an edge appear at a given time or not.

$$I^{SN_M} = mI^m + tI^t + te$$

where the first element is the cost of encoding the edges, the second element encodes the timestamps, and the last encode the bits of the matrix.

In the second snapshot representation, each snapshot is represented as a list of pair of nodes, and timestamps are added at the start of every snapshot. This representation is equivalent to representing each snapshot as an edge list.

$$I^{SN_E} = eI^m + tI^t + tI^m$$

Where the first element is the cost of encoding the edges, the second element encodes timestamps, and the last encode stop section at the end of each snapshot.

Interval Graph Encoding. For this representation, we need to introduce a new property: the encoding length won't depend on the total number of events e , but on the total number of intervals i .

As explained in Sect. 2, an interval of edge existence corresponds to a period of time during which all possible observations if this edge are present. For instance, if observations occur every year, and the edge e is observed in 2010,2011,2012 and 2013 but not in 2009 and 2014, then the four observations between 2010 and 2013 can be replaced by a single interval [2010,2014].

As a consequence, we also define $t' \leq t$ the total number of different intervals endpoints, and $I^{t'} = \log_2(t')$.

$$I^{IG} = mI^m + 2iI^{t'} + mI^{t'}$$

Where the first element is the cost of encoding the edges, the second element encodes the intervals, and the last encode stop sections as for link streams.

4 Experiments

To validate the relevance of our approach, we experiment with synthetic and real networks. In each experiment, we test with the original temporal resolution (on the left of figures), and then explore how aggregating at coarser temporal scales affects encoding costs. To create those aggregated version, we use non-overlapping sliding time-windows. To every unique time period, we associate an unweighted cumulative graph, such as an edge exists between two nodes of this graph if there is at least one interaction between those two nodes during the corresponding period.

All experiments can be checked and reproduced using an on-line notebook³.

4.1 Synthetic Networks

We generate three types of synthetic dynamic networks and compute encoding length on them for our four models. Results are synthesized in Fig. 2.

³ https://colab.research.google.com/github/Yquetzal/tnetwork/blob/master/article_encoding.ipynb.

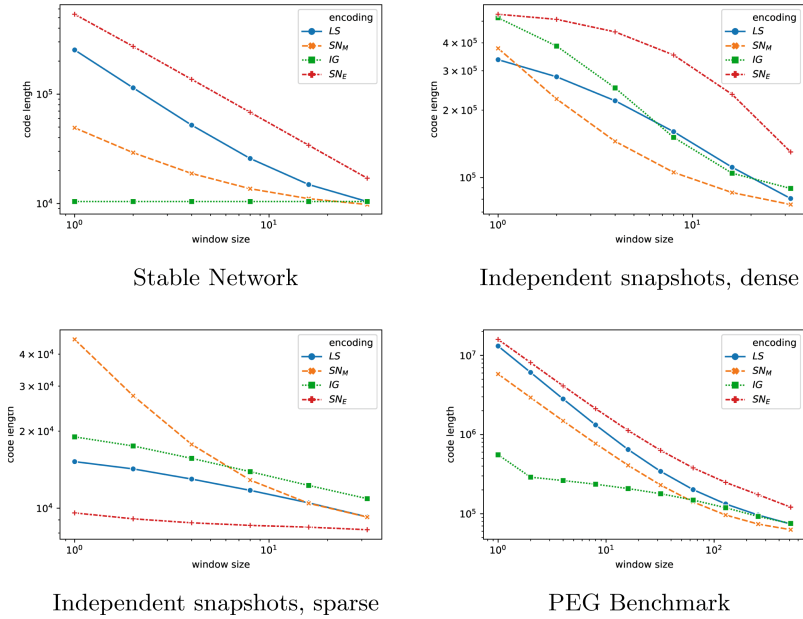


Fig. 2. Code length for synthetic graphs. We observe that the most efficient network representation depends strongly on the properties of networks.

Stable Network. In the first experiment, we generate a single static Erdős-Rényi random graph with 100 nodes and 640 edges and create a dynamic network composed of 64 identical snapshots. The properties of this temporal graph is therefore $(n = 100, m = 640, t = 64, e = 40960)$. We can observe that the Interval Graph approach allows the most efficient encoding when there are at least two snapshots. This matches our expectation, since the network is perfectly *stable* (long intervals). Link streams and SN_E are the least efficient, because they repeat uselessly timestamps and edges, respectively.

Independent Snapshots, Dense. In the second experiment, we generate a network with $t = 64$ snapshots, each corresponding to a static Erdős-Rényi random graph with the same property as previously (100 nodes, 640 edges). We observe in Fig. 2 that the link stream representation is the most efficient without aggregation, but the matrix snapshot representation becomes more efficient as soon as we aggregate. This is due to the important density: with a total of $e = 40960$ observed interaction for 4950 possible node pairs, most pairs are repeated multiple times.

Independent Snapshots, Sparse. In this experiment, we generate a network with $t = 64$ snapshots, each corresponding to a static Erdős-Rényi random graph with 100 nodes and 10 edges. The number of observations ($e = 640$) is now equivalent to the number of different edges m observed in the first experiment. We observe that the SN_E representation is now the most efficient, which is coherent with our expectation, since this representation is efficient when edge repetitions are rare.

Progressively Evolving Graph Benchmark. In the last experiment, we use an existing benchmark to generate a dynamic network. Published in [4], this benchmark generates progressively evolving graphs with changing community structures. We use the standard generator used in [4], yielding a temporal network with $n = 92, m = 4169, e = 2007930, t = 1961$. We observe that the interval graph representation is by far the most efficient on this network, which is due to the progressively evolving nature of the graphs: edges present in a period tend to be also present in the next. However, because there is a long term evolution and some random noise, the matrix representation SN_M is not efficient.

4.2 Experiments with Real Networks

In this section, we apply the same procedure on temporal networks corresponding to real datasets. We summarize information on those networks in Table 1. Three networks come from the SocioPatterns project [1], SP-HS (High-School), SP-Hosp (Hospital) and SP-PS (Primary School). As already mentioned in Sect. 2, they correspond to interactions collected between individuals every 20s. ENRON is a dataset of emails sent between employees. Temporal information is available at the level of the minute, over a period of about 3 years. Primates is a dataset of social interactions between primates, collected over 19 periods. GOT is a network of interactions between characters of a TV series (Game of Thrones) over several seasons, originally aggregated every 10 scenes.

Table 1. Summary of real networks properties. n: number of nodes. m: number of different edges. e: number of interactions. t: number of timesteps. e/t: average number of observations per timesteps. e/m: average number of observation per edge. e/m/t: average probability to observe an existing edge at a given step.

Network	n	m	e	t	e/t	e/m	e/m/t(%)
SP-HS [7]	180	2220	45047	11273	4	20.29	0.18
SP-Hosp [16]	75	1139	32424	9453	3.4	28.4	0.3
SP-PS [15]	242	8317	125773	3100	40.6	15.1	0.49
ENRON [14]	150	1526	24694	14832	1.7	16.2	0.11
Primates[14]	25	280	1340	19	70.5	4.8	0.25
GOT [2]	338	939	20011	1031	19.4	21.3	2.07

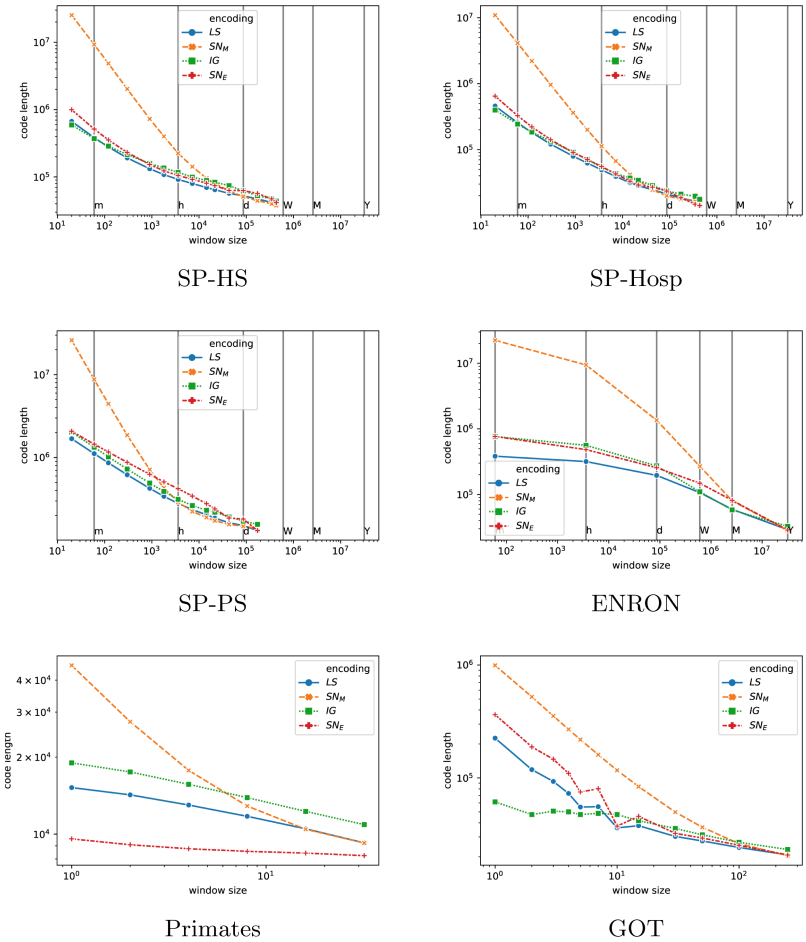


Fig. 3. Code length for synthetic graphs. We observe that the most efficient network representation depends strongly on the properties of networks.

All datasets are public, and available either through their original paper or through the network data repository [14], as reported in Table 1.

Results of experiments are reported in Fig. 3. When relevant, we indicate with vertical lines typical temporal scales (respectively, minute, hour, day, week, month, year).

The three sociopattern datasets seem to have similar profiles overall. For High School and Hospital datasets, Interval Graphs is the most efficient representation for the original timescale, while Link Streams become more efficient if we aggregate every 5 min, approximately. From this observation, we can infer that interactions tend to be maintained typically for no more than a few minutes. Each interval of observation thus becomes a single observation when aggregating, more efficiently represented as a Link Stream. For the Primary

School, the Link Stream representation is the most efficient even for the original data, either because the data is more noisy or because there are more singleton observations.

Link Stream is also the most efficient representation for ENRON dataset, as expected due to the nature of the dataset: each email is stamped with the exact minute it was sent, and it is rather unlikely that emails sent on a particular minute form a well-defined graph, or that several emails are sent between the same individuals in successive minutes. Only when aggregating at a scale of weeks or months is the Interval Graph representation the most appropriate, and when aggregating every year a snapshot representation becomes relevant.

In the primate dataset on the contrary, the snapshot representation is the most appropriate: each timestamp correspond to a well formed graph, and relations are usually not stables from one snapshot to the next.

Finally, for the Game of Thrones dataset, Interval graphs seems to be clearly the most appropriate for the original data. However, by having a second look at the data, we realized that this is due to the way the dataset is provided. A smoothing window is used, and for each sequence of 10 scenes, the same average network is provided 10 times, i.e., snapshots 1–10 corresponds to the first 10 scenes and are identical. Using our method, we observe that if we aggregate every 10 scenes, thus removing this bias, the link stream approach becomes the most efficient.

5 Conclusion

In this article, we have introduced a method to choose an appropriate representation for a temporal network, and validated its relevance on synthetic and real networks.

This method is implemented in the `tnetwork` python library to automatically select the right representation when loading a file, and to store more efficiently temporal networks.

Beyond these practical aspects, choosing the most appropriate representation is essential to know how to handle a network and which algorithms or methods can be applied to it. In future works, we wish to analyze further how to select an appropriate aggregation scale to transform efficiently interaction datasets – which seem to be the most frequent in real data – into stable networks that can be analyzed as interval graphs or snapshots, while losing as little temporal information as possible.

References

1. Barrat, A., Cattuto, C., Colizza, V., Gesualdo, F., Isella, L., Pandolfi, E., Pinton, J.F., Ravà, L., Rizzo, C., Romano, M., et al.: Empirical temporal networks of face-to-face human interactions. *Eur. Phys. J. Spec. Topics* **222**(6), 1295–1309 (2013)

2. Bost, X., Labatut, V., Gueye, S., Linares, G.: Narrative smoothing: dynamic conversational network for the analysis of TV series plots. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 1111–1118. IEEE (2016)
3. Cazabet, R., Amblard, F., Hanachi, C.: Detection of overlapping communities in dynamical social networks. In: 2010 IEEE Second International Conference on Social Computing, pp. 309–314. IEEE (2010)
4. Cazabet, R., Boudebza, S., Rossetti, G.: Evaluating community detection algorithms for progressively evolving graphs. *J. Complex Netw.* (2020)
5. Remy Cazabet, R., Rossetti, G.: Challenges in community discovery on temporal networks. In: *Temporal Network Theory*, pp. 181–197. Springer, Heidelberg (2019)
6. Coscia, M., Rossetti, G., Giannotti, F., Pedreschi, D.: Demon: a local-first discovery method for overlapping communities. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 615–623 (2012)
7. Fournet, J., Barrat, A.: Contact patterns among high school students. *PLoS ONE* **9**(9), e107878 (2014)
8. Gauvin, L., Génois, M., Karsai, M., Kivelä, M., Takaguchi, T., Valdano, E., Vestergaard, C.L.: Randomized reference models for temporal networks. *arXiv preprint arXiv:1806.04032* (2018)
9. Grünwald, P.D., Grunwald, A.: *The Minimum Description Length Principle*. MIT Press, Cambridge (2007)
10. Holme, P., Saramäki, J.: Temporal networks. *Phys. Rep.* **519**(3), 97–125 (2012)
11. Latapy, M., Viard, T., Magnien, C.: Stream graphs and link streams for the modeling of interactions over time. *Soc. Netw. Anal. Mining* **8**(1), 61 (2018)
12. Matias, C., Rebafka, T., Villers, F.: A semiparametric extension of the stochastic block model for longitudinal networks. *Biometrika* **105**(3), 665–680 (2018)
13. Mucha, P.J., Richardson, T., Macon, K., Porter, M.A., Onnela, J.-P.: Community structure in time-dependent, multiscale, and multiplex networks. *Science* **328**(5980), 876–878 (2010)
14. Rossi, R.A., Ahmed, N.K.: The network data repository with interactive graph analytics and visualization. In: *AAAI* (2015)
15. Stehlé, J., Voirin, N., Barrat, A., Cattuto, C., Isella, L., Pinton, J.-F., Quaghiotto, M., Van den Broeck, W., Régis, C., Lina, B., Vanhems, P.: High-resolution measurements of face-to-face contact patterns in a primary school. *PLOS ONE* **6**(8), e23176 (2011)
16. Vanhems, P., Barrat, A., Cattuto, C., Pinton, J.F., Khanafer, N., Régis, C., Kim, B.A., Comte, B., Voirin, N.: Estimating potential infection transmission routes in hospital wards using wearable proximity sensors. *PLoS ONE* **8**(9), e73970 (2013)
17. Viard, T., Latapy, M., Magnien, C.: Computing maximal cliques in link streams. *Theoret. Comput. Sci.* **609**, 245–252 (2016)



Effect of Nonisochronicity on the Chimera States in Coupled Nonlinear Oscillators

K. Premalatha¹(✉), V. K. Chandrasekar², M. Senthilvelan³, R. Amuda¹,
and M. Lakshmanan³

¹ Centre for Nonlinear Dynamics, Department of Physics, PSG College of Technology,
Coimbatore 641 004, Tamil Nadu, India

snkpremalatha@gmail.com

² Centre for Nonlinear Science and Engineering, School of Electrical and Electronics
Engineering, SASTRA University, Thanjavur 613 401, Tamilnadu, India

³ Centre for Nonlinear Dynamics, School of Physics, Bharathidasan University,
Tiruchirappalli 620 024, Tamil Nadu, India

Abstract. We investigate the conditions which enable one to show the phenomenon of swing of synchronized states via amplitude chimera states in non-locally coupled systems with symmetry breaking with coupled Stuart-Landau oscillators as an example. Chimera states represent the spatio-temporal patterns coexisting with synchronized and desynchronized behaviour in coupled identical oscillators. We identify that the radius of non-local interaction and the non-isochronicity in the system also play an important role in the observation of such states. The system shows such notable property neither for smaller values of non-isochronicity nor for higher values. We also carefully study the occurrence of different transition routes to recently observed dynamical state called chimera death while varying the strength of nonisochronicity parameter.

Keywords: Chimera states · Coupled oscillators · Nonisochronicity parameter

1 Introduction

Chimera states are intriguing spatiotemporal patterns coexisting with synchronized and desynchronized oscillations and it has brought out considerable attention towards the study of coupled networks with nonlocal topology. It has been realized that nonlocal coupling plays a crucial role in inducing chimera states [1–6]. Recently, such coexistence behavior has also been addressed in global coupling [7] and local coupling [8–10]. They have also been observed in maps [11], complex networks [12], time discrete and continuous chaotic systems [13]. This state has also been observed experimentally in mechanical oscillators with metronomes [14], coupled chemical oscillators [15], coupled electronic oscillators [16], oscillators with more than one populations [17–21], time varying networks [22] and also in optical coupled map lattices realized by liquid crystal light modulators [23]. These observations of chimera states have helped to explain various phenomena that occur in practice, such as uni-hemispheric sleep [24], power distribution in networks [25], and bump states in neural networks [26].

Another emergent phenomenon in nonlocally connected identical oscillators is the appearance of chimera death in the presence of symmetry breaking coupling term as was observed by Anna Zakharova et al. [27]. In the chimera death state, the oscillators in the network partition into two coexisting domains, where in one domain neighboring nodes occupy the same branch of the inhomogeneous steady state (spatially coherent oscillation death) while in the other domain neighboring nodes are randomly distributed among the different branches of the inhomogeneous steady state (spatially incoherent oscillation death). In a recent work, the present authors have reported that the chimera death can be induced in a network of oscillators with global coupling also leading to multi-cluster chimera death [28] and structural changes in the chimera death region under nonlocal coupling interaction [29].

From a different point of view, Daido and Nakanishi in [30] have observed the interesting phenomenon of swing of synchronized states, while studying the inhomogeneity induced by the introduction of diffusive coupling in globally coupled oscillators. They found that the synchronized state which has been destabilized because of the increase in coupling strength is found to be restabilized for further raise in the coupling strength. The diffusion in globally coupled systems induces the synchronized state to be mediated by the so called cluster states. Inspired by the work of Daido and Nakanishi [30], we have investigated the question whether global coupling is a prerequisite for the above type of swing phenomenon. Interestingly, in the present work we observe the same phenomenon, namely synchronized state being mediated through amplitude chimera states in nonlocally coupled systems with symmetry breaking introduced therein.

In this article, we study the behavior of nonlinear oscillators that are coupled nonlocally with symmetry breaking form. We observe the basic feature that the nonisochronicity in the system is a key ingredient in realizing the synchronized state mediated through the amplitude chimera states and that we cannot observe such a phenomenon when there is an absence of nonisochronicity in the system. The other crucial contributor in inducing sway of synchronized states is the coupling radius of non-local interaction. Here the presence of nonlocal coupling in the system makes the amplitude chimera states to mediate the synchronized states. We illustrate the above results with one of the ubiquitous interacting models, namely the coupled Stuart-Landau (SL) oscillators. In order to explore the phenomenon of swing in the synchronized state which is mediated through the amplitude chimera state, we use the characteristic measure that is strength of incoherence [31]. The diverse transition routes of the system towards the dynamical state namely chimera death (due to symmetry breaking in the coupling) for different coupling radius and nonisochronicity parameter have also been obtained.

2 Swing-By Mechanism and Chimera Death in Coupled Stuart-Landau Oscillators Under Nonlocal Coupling with Symmetry Breaking

We consider an array of nonlocally coupled identical Stuart-Landau oscillators with symmetry breaking in the coupling, whose dynamics can be represented by the following set of equations,

$$\dot{z}_j = z_j - (1 - ic)|z_j|^2 z_j + \frac{\varepsilon}{2P} \sum_{k=N-P}^{N+P} (Re[z_k] - Re[z_j]), \quad (1)$$

where $z_j = x_j + iy_j$, $j = 1, 2, 3, \dots, N$. All the indexes in Eq. (1) are regarded as modulo N . Here c is the nonisochronicity parameter and N is the total number of oscillators. The nonlocal coupling in the system is controlled by the coupling strength (ε) and the coupling range ($r = \frac{P}{N}$), where P corresponds to the number of nearest neighbors in both directions (or coupling radius). Here, we have introduced the coupling only in the real parts of the complex amplitude, and so this coupling introduces a symmetry breaking in the system. This symmetry breaking is important here as we find that the absence of symmetry breaking cannot induce the undulation in synchronization as we prove in the next section.

In our simulations, we choose the number of oscillators N to be equal to 100 and in order to solve the Eq. (1), we use the fourth order Runge-Kutta method with time step 0.01 and with symmetric initial conditions between -1 and $+1$ which is necessary for the occurrence of oscillation death state.

2.1 Characterization of Chimera and Other Collective States

In this section we use the quantitative measure strength of incoherence [31] to characterize the different dynamical states such as desynchronized, chimera and synchronized states, which are described below.

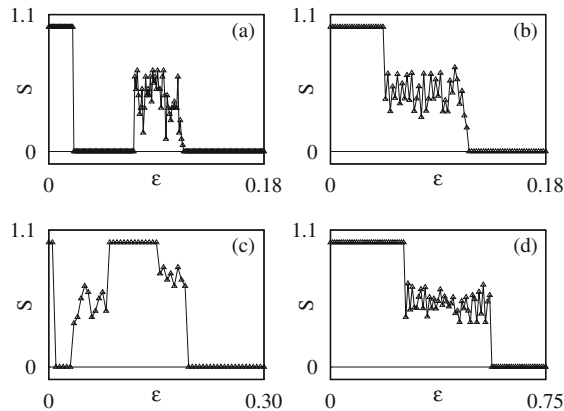


Fig. 1. (a) Strength of incoherence (S) of the system (1) for different values of ε (a) for $c = 3$ and $P = 10$, (b) $c = 3$ and $P = 25$, (c) $P = 10$ with $c = 4.5$ and (d) for $P = 10$, $c = 7$, respectively.

Characterization in Terms of Strength of Incoherence. In order to check whether the system exhibits the swing of synchronization mechanism in any region of the parametric space of this nonlocally coupled system, we vary the coupling range r , or equivalently P , and the nonisochronicity parameter c , and observe the dynamical behavior of the

system. In order to know the nature of dynamical states in more detail, we look at the strength of incoherence of the system which was introduced recently by Gopal, Venkatesan and two of the present authors [31] that will help us to detect interesting collective dynamical states such as the chimera state. It is defined as

$$S = 1 - \frac{\sum_{m=1}^M s_m}{M}, \quad s_m = \Theta(\delta - \sigma(m)), \tag{2}$$

where $\sigma(m) = \langle (\frac{1}{n} \sum_{j=n(m-1)+1}^{mn} |z_j - \bar{z}_j|^2)^{1/2} \rangle_t$, δ is the threshold value which is small and Θ is the Heaviside step function. The angular bracket $\langle \dots \rangle_t$ denotes the average over time. Thus the strength of incoherence measures the amount of spatial incoherence present in the system which is zero for the spatially coherent/synchronized state. It has the maximum value, that is $S = 1$, for the completely incoherent/desynchronized state and has intermediate values between 0 and 1 for chimera states and cluster states. Now using the above strength of incoherence S , we identify the different dynamical regions which the system passes through while the coupling radius and nonisochronicity parameter are varied. For this purpose, in Fig. 1, we demonstrate the behavior of the strength of incoherence (S) for the variables x_j with respect to the coupling strength ϵ for coupling radius $P = 10$ with $c = 3$. One finds that initially all the oscillators are desynchronized, where the value of S is found to be maximum. However, in the region $0.01 \leq \epsilon < 0.021$ the system of oscillators attain synchronized state where ($S = 0$). On increasing the coupling strength, in the region $0.072 < \epsilon < 0.11$, S oscillates between 0 to 1, implying that the states correspond to a chimera state. By increasing ϵ beyond 0.11, S is found to be zero which confirms the synchronization among the oscillators. Thus, in this case, we can observe a recurrence of synchronization, where the synchronization in the system disappears with the increase of ϵ but with further increase of ϵ synchronized states again reemerge. The above analysis shows that the swing in synchronization in the system is mediated by the chimera state and the corresponding transition route is represented as **desynchronization** \rightarrow **synchronization** \rightarrow **chimera state** \rightarrow **synchronization**.

Now increasing the coupling radius to $P = 25$, we cannot observe this type of sway in synchronization, which is also shown in Fig. 1(b). Initially the states are desynchronized where $S = 1$ in the region $\epsilon < 0.045$ and S takes values in between 0 to 1 in the region $0.046 < \epsilon < 0.11$ indicating the presence of chimera states. For $\epsilon > 0.11$ the states are synchronized where S takes the value zero. Thus, we can observe here the absence of recurrence of the synchronized state for large values of nonlocal interaction. It follows the route **desynchronization** \rightarrow **chimera states** \rightarrow **synchronization**. Thus, from the above Figs. 1(a) and 1(b) we find that the swing in synchronization occurs for small values of coupling radius (or strength of nonlocal interaction). Now we check whether this type of undulation of synchronization occurs for all values of c .

For this purpose, we fix $P = 10$ and find S for different values of c . The calculated results show that the swing mechanism in synchronization occurs neither for large values of c nor for smaller values of c but can be observed for the window $2.7 \leq c \leq 5.2$. To illustrate this, we have plotted the strength of incoherence (S) of the system for two different values of c , namely $c = 4.5$ and $c = 7.0$ in Fig. 1(c) and (d), respectively. At $c = 4.5$ (from Fig. 1(c)), one can observe that for smaller values of ϵ ($\epsilon < 0.02$) S is

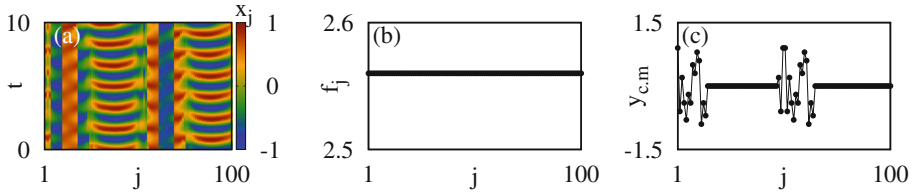


Fig. 2. (a) Snap shot of the variable x_j corresponding to amplitude chimera state with $P = 10$, $c = 3.0$ and $\varepsilon = 0.04$. (b) Frequency of the oscillators in the system for amplitude chimera state, (c) corresponding center of mass $y_{c,m}$ averaged over one period of each oscillator for the variable y_j .

found to be unity which represents the desynchronization among the oscillators and by increasing the coupling S reaches the value zero where the oscillators are in synchronization. Further for the values of ε between 0.036 and 0.08, S has non-zero values, but below one. We find the occurrence of chimera states in the region $0.036 < \varepsilon < 0.082$ and $0.15 < \varepsilon < 0.19$ (as S takes the values between 0 and 1 for these values of ε). For the intermediate values of ε ($0.082 < \varepsilon < 0.15$) S is found to have unit value indicating that the state is desynchronized. S is found to reach the minimum value ($S = 0$) by further increase of ε . Thus for the values $\varepsilon > 0.19$ the oscillators return back to the synchronized state. The above analysis shows that the swing in synchronization in the system is mediated by the desynchronized state in addition to the chimera states. More specifically the states of the oscillators in this case follow the transition route as *desynchronization* \rightarrow *synchronization* \rightarrow *chimera state* \rightarrow *desynchronization* \rightarrow *chimera state* \rightarrow *synchronization*. Now let us look whether this type of reappearance occurs for higher values of c also.

(iv) The obtained results for $c = 7.0$ are shown in Fig. 1(d) which indicates the absence of the above type of synchronized state. In this case, we can observe that S has the maximum value for small values of ε ($\varepsilon < 0.25$). S takes the value between 0 to 1 for $0.25 < \varepsilon < 0.550$ corresponding to a chimera state. Finally for $\varepsilon > 0.550$, S decreases to zero corresponding to desynchronized state. The transition route in the present case is then as follows: *desynchronization* \rightarrow *chimera state* \rightarrow *synchronization*. Hence the large value of nonisochronicity leads to the absence of swing in synchronized state. Thus, for $P = 10$, we have the phenomenon of swing by in synchronization for the values of c between $2.7 < c < 5.2$.

Nature of Chimera States and Chimera Death States. In order to know the characteristic nature of chimera states more clearly, we present the space-time plot and frequency profiles corresponding to the system in the chimera state region. Figure 2(a) shows that there exists fluctuation in amplitudes of the oscillations (Fig. 2(b)) of the desynchronized oscillators for $p = 10$, $c = 3.0$ and $\varepsilon = 0.8$. But Fig. 2(b) shows that the frequency of the oscillators are identical. This is also confirmed through by calculating center of mass of the oscillators in Fig. 2(c) which clearly illustrate that the synchronized oscillators are oscillating periodically with origin as the center of rotation while the incoherent oscillators are oscillating periodically with different amplitudes and with

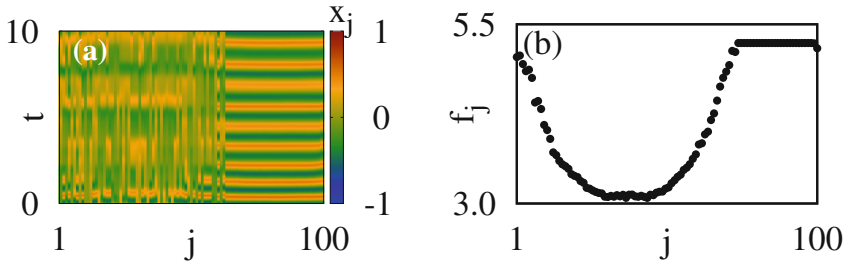


Fig. 3. (a) Snap shot of the variables x_j corresponding to frequency chimera state with $P = 40$, $c = 7.0$ and $\varepsilon = 0.08$. (b) Frequency of the oscillators in the frequency chimera state shown in (a).

a shifted center of rotation from the origin [27]. Hence we confirm that the swing of synchronized states occurs through the amplitude chimera states. Interestingly, for larger strength of nonlocal interaction, we observe a change in the behavior of the chimera states. For the case $P = 40$, $c = 7.0$ and $\varepsilon = 0.08$, Fig. 3 shows that in addition to the fluctuations in the amplitude, there are fluctuations in their frequencies also. The frequency profile Fig. 3 (b) of the oscillators shows that there are two groups of oscillators: the oscillators from 1 to 73 belong to the first group, and they have different frequencies, thereby showing the presence of spatial incoherence among these oscillators. The other group is made up of the oscillators from 74 to 100, which are all found to have the same frequency and are synchronized. This type of chimera state can be designated as frequency chimera state.

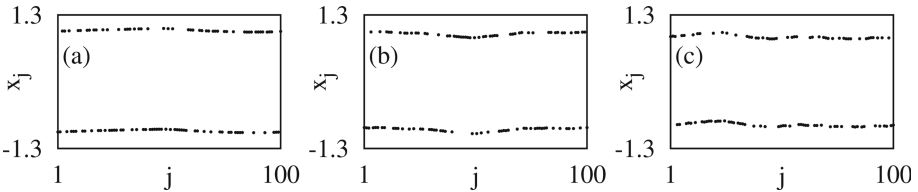


Fig. 4. Snap shot of the variables x_j corresponding to chimera death state with $P = 10$ and $\varepsilon = 0.08$: (a) for $c = 0$, (b) for $c = 3$, (c) for $c = 6$.

Further we note that the presence of symmetry breaking term in the coupling causes the system to transit to a new dynamical state called chimera death state for large values of coupling strength. It has the combined properties of chimera and oscillation death (OD). The population of identical oscillators splits into two coexisting domains: (i) spatially coherent oscillation death (neighboring oscillators populate in the same branch of inhomogeneous steady state either as $x^{(1)}$ or $x^{(2)}$) (ii) spatially incoherent oscillation death (population of neighboring oscillators are completely random between $x^{(1)}$ and $x^{(2)}$), which is clearly shown in Fig. 4 for three different value of nonisochronicity parameter. Other parameter values are fixed as $P = 10$ and $\varepsilon = 0.8$. In this steady

state all the oscillators in the array are distributed uniformly either in the lower branch or in the upper branch for $c = 0$ as shown in Fig. 4(a). Next on increasing the value of the nonisochronicity parameter to $c = 3$, as seen in Fig. 4(b), we observe that an increase in disorder in the distribution of inhomogeneous steady states. On increasing nonisochronicity parameter further ($c = 6$), one finds a further increase in disorder in distribution of inhomogeneous steady states as in Fig. 4(c). Thus we conclude that increase of strength of nonisochronicity parameter leads to an increase in the distribution of inhomogeneous steady states. To give a concrete idea about the different dynamical states with transition routes, we extend our study with phase diagram in the next sub-section.

2.2 Collective States in the (ε, c) Parameter Space

In order to give a global picture of the different transition routes to chimera death in the system more clearly, we present a phase diagram of the system for $P = 10$ in Fig. 5 (a). It shows that the system for finite values of c ($c < 2.7$) is found to be synchronized by the increase of ε and the symmetry breaking present in the system causes chimera death for larger ε . An increase in c (to $c \approx 3$) causes the synchronized state (that appears through the increase of ε) to be destabilized for a further increase of ε giving rise to amplitude chimera state and further increase of ε restabilizes the synchronized state. This indicates the presence of swing in the synchronization phenomenon. As mentioned earlier, the increase in c suppresses in the amount of spatial coherence in the system for lower values of ε . Thus, the increase in c causes the transition of chimera state to desynchronized state. For example, for $c \approx 5$, the transition route to chimera death can be identified as *desynchronization* \rightarrow *synchronization* \rightarrow *amplitude chimera* \rightarrow *desynchronization* \rightarrow *amplitude chimera* \rightarrow *synchronization* \rightarrow *chimera death*. For values of $c > 5.3$, the synchronization for lower values of ε completely loses its stability and there is a cessation of swing in synchronization phenomenon in the system. Also on the whole, we can observe that an increase in c weakens spatial coherence of the system for lower ε and strengthens the spatial coherence for higher ε . We can find the multistability between the stable amplitude chimera state and synchronized state in region-AC. That is, we can observe the stable amplitude chimera state for specific choice of initial condition. In that region, we can also find that the synchronized solution coexist for the initial condition near the synchronized solution. We can find the multistability between the chimera death state and synchronized state in region-CD. In region-CD, we can observe the chimera death state for specific choice of initial condition. In that region, we can also find that the synchronized solution coexist for the initial condition near the synchronized solution.

Similarly, Fig. 5 (b) shows the phase diagram of the system in the (ε, c) parametric space for $P = 40$. It shows that an increase in the radius of nonlocal interaction also suppresses the synchronized state which is mediated through the amplitude chimera state. Here, we can find that for smaller values of c the system shows direct transition from desynchronized state to chimera death, whereas an increase in c in the region $4 \leq c < 5.0$ causes the amplitude chimera state to mediate this transition. As mentioned earlier, further increase in c causes the stabilization of synchronized state for higher ε . Interestingly in this case, we can observe that increasing nonisochronicity parameter

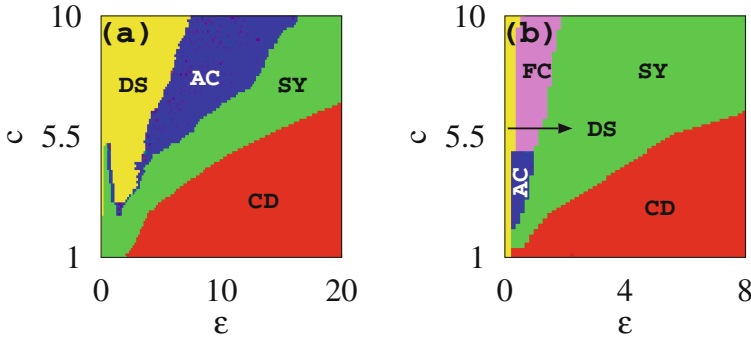


Fig. 5. (a) (Color online) Phase diagrams of the system (1) for (a) $P = 10$, (b) $P = 40$. (This figure is reproduced from [28]). SY (Green color), DS (yellow), AC (blue), FC (violet), CD (red) regions represent synchronized state, desynchronized state, amplitude chimera state, frequency chimera state, and chimera death state (CD), respectively.

increases disorder in the system. At $c = 7.0$, a variation in the coupling strength causes the system to transit from a desynchronized state to synchronized state via frequency chimera instead of amplitude chimera due to the increase of nonisochronicity parameter which induces the disorder in the dynamical states. The diverse transition routes to chimera death is briefly explained in the Table 1.

Table 1. Different transition routes: DS \rightarrow Desynchronized states, SY \rightarrow synchronized states, AC \rightarrow amplitude chimera states, FC \rightarrow frequency chimera states, CD \rightarrow chimera death states.

S. No	Coupling schemes	Mechanism
1	Nonlocal coupling $P = 10$	(i) DS \rightarrow SY \rightarrow CD (ii) DS \rightarrow SY \rightarrow AC \rightarrow SY \rightarrow CD (iii) DS \rightarrow SY \rightarrow AC \rightarrow DS \rightarrow AC \rightarrow SY \rightarrow CD (iv) DS \rightarrow AC \rightarrow SY \rightarrow CD
2	$P = 40$	(v) DS \rightarrow CD (vi) DS \rightarrow AC \rightarrow SY \rightarrow CD (vii) DS \rightarrow FC \rightarrow SY \rightarrow CD

3 Conclusion

In summary, we have investigated the occurrence of synchronized oscillations via amplitude chimera states in nonlocally coupled systems with symmetry breaking interaction. We illustrated the roles of nonlocal interaction and the strength of nonisochronicity in inducing such type of synchronized states. Our results show that in the nonlocally coupled system with symmetry breaking, the occurrence of characteristic feature in synchronization is observed for smaller values of nonlocal interaction.

Even with small radius of nonlocal interaction, the system shows such notable property neither for smaller values of nonisochronicity nor for higher values of nonisochronicity. The swing of synchronized state initially follows the route that is defined as *synchronization* \rightarrow *amplitude chimera* \rightarrow *synchronization* and the increase in the nonisochronicity causes the synchronized state to be mediated by the desynchronized state along with the amplitude chimera state, where the transition route can be defined as *synchronization* \rightarrow *amplitude chimera* \rightarrow *desynchronization* \rightarrow *amplitude chimera* \rightarrow *synchronization*.

Another interesting case of these nonlocally coupled systems for higher radius of nonlocal interaction P (in which case such peculiarity of synchronization disappears) is that the presence of frequency chimera state due to disorder introduced by the nonisochronicity parameter. We also carefully study the occurrence of different transition routes to recently observed dynamical state called chimera death while varying the strength of nonisochronicity parameter. Since, we have studied the model of nonlocally coupled Stuart-Landau oscillators, which has wide practical applications in physical, chemical and biological phenomena, this study may help to control the chimera states which appear in various areas. As an example, chimera states in power distribution networks may lead to blackouts due to coexistence of desynchronized state with synchronized state. Controlling chimera may lead to maintain the stable distribution of power supply.

Acknowledgement. The work of K. P has been supported by the DST-SERB, Government of India, for providing financial support through National Post Doctoral Fellowship under the grant No. PDF/2018/000783. The work of VKC forms part of a research projectsponsored by CSIR Project under Grant No. 03(1444)/18/ EMR-II. The work of M. S. forms part of a research project sponsored by the Council of Scientific and Industrial Research (CSIR), Government of India under the grant number 03/1397/17/EMR-II. M.L.wishes to thank the Department of Science and Technology for the award of a SERB Distinguished Fellowship underGrant No. SB/DF/04/2017.

References

1. Kuramoto, Y., Battogtokh, D.: Coexistence of coherence and incoherence in nonlocally coupled phase oscillators. *Nonlinear Phenom. Complex Syst.* **5**, 380–385 (2002). <http://www.jnpcs.org/online/vol2002/v5no4/v5no4p380.pdf>
2. Abrams, D.M., Strogatz, S.H.: Chimera states for coupled oscillators. *Phys. Rev. Lett.* **93**, 174102 (2004). <https://doi.org/10.1103/PhysRevLett.93.174102>
3. Abrams, D.M., Strogatz, S.H.: Chimera states in a ring of nonlocally coupled oscillators. *Int. J. Bif. Choas.* **16**(1), 21 (2006). <https://doi.org/10.1142/S0218127406014551>
4. Abrams, D.M., Mirollo, R., Strogatz, S.H., Wiley, D.A.: Solvable model for chimera states of coupled oscillators. *Phys. Rev. Lett.* **101**, 084103 (2008). <https://doi.org/10.1103/PhysRevLett.101.084103>
5. Shima, S.I., Kuramoto, Y.: Rotating spiral waves with phase-randomized core in nonlocally coupled oscillators. *Phys. Rev. E* **69**, 036213 (2004). <https://doi.org/10.1103/PhysRevE.69.036213>
6. Sheeba, J.H., Chandrasekar, V.K., Lakshmanan, M.: Chimera and globally clustered chimera: impact of time delay. *Phys. Rev. E* **81**, 046203 (2010). <https://doi.org/10.1103/PhysRevE.81.046203>

7. Sethia, G.C., Sen, A.: Chimera states: the existence criteria revisited. *Phys. Rev. Lett.* **112**, 144101 (2014). <https://doi.org/10.1103/PhysRevLett.112.144101>
8. Laing, C.R.: Chimeras in networks with purely local coupling. *Phys. Rev. E* **92**, 050904(R) (2015). <https://doi.org/10.1103/PhysRevE.92.050904>
9. Bera, B.K., Ghosh, D., Lakshmanan, M.: Chimera states in bursting neurons. *Phys. Rev. E* **93**, 012205 (2016). <https://doi.org/10.1103/PhysRevE.93.012205>
10. Premalatha, K., Chandrasekar, V.K., Senthilvelan, M., Lakshmanan, M.: Stable amplitude chimera states in a network of locally coupled Stuart-Landau oscillators. *Chaos* **28**, 033110 (2018). <https://doi.org/10.1063/1.5006454>
11. Omelchenko, I., Maistrenko, Y., Hövel, P., Schöll, E.: Loss of coherence in dynamical networks: spatial chaos and chimera states. *Phys. Rev. Lett.* **106**, 234102 (2011). <https://doi.org/10.1103/PhysRevLett.106.234102>
12. Zhu, Y., Zheng, Z., Yang, J.: Chimera states on complex networks. *Phys. Rev. E* **89**, 022914 (2014). <https://doi.org/10.1103/PhysRevE.89.022914>
13. Omelchenko, I., Riemenschneider, B., Hövel, P., Maistrenko, Y.L., Schöll, E.: Transition from spatial coherence to incoherence in coupled chaotic systems. *Phys. Rev. E* **85**, 026212 (2012). <https://doi.org/10.1103/PhysRevE.85.026212>
14. Martens, E.A., Thutupalli, S., Fourrière, A., Hal-latschek, O.: Chimera states in mechanical oscillator networks. *Proc. Natl. Acad. Sci.* **110**, 10563 (2013). <https://doi.org/10.1073/pnas.1302880110>
15. Tinsley, M.R., Nkomo, S., Showalter, K.: Chimera and phase-cluster states in populations of coupled chemical oscillators. *Nat. Phys.* **8**, 662 (2012). <https://doi.org/10.1038/nphys2371>
16. Gambuzza, L.V., Buscarino, A., Chessa, S., Fortuna, L., Meucci, R., Frasca, M.: Experimental investigation of chimera states with quiescent and synchronous domains in coupled electronic oscillators. *Phys. Rev. E* **90**, 032905 (2014). <https://doi.org/10.1103/PhysRevE.90.032905>
17. Montbrio, T.E., Kurths, J., Blasius, B.: Synchronization of two interacting populations of oscillators. *Phys. Rev. E* **70**, 056125 (2004). <https://doi.org/10.1103/PhysRevE.70.056125>
18. Pikovsky, A., Rosenblum, M.: Partially integrable dynamics of hierarchical populations of coupled oscillators. *Phys. Rev. Lett.* **101**, 264103 (2008). <https://doi.org/10.1103/PhysRevLett.101.264103>
19. Martens, E.A., Panaggio, M.J., Abrams, D.M.: Basins of attraction for chimera states. *New J. Phys.* **18**, 022002 (2016). <https://doi.org/10.1088/1367-2630/18/2/022002>
20. Premalatha, K., Chandrasekar, V.K., Senthilvelan, M., Lakshmanan, M.: Imperfectly synchronized states and chimera states in two interacting populations of nonlocally coupled Stuart-Landau oscillators. *Phys. Rev. E* **94**, 012311 (2016). <https://doi.org/10.1103/PhysRevE.94.012311>
21. Premalatha, K., Chandrasekar, V.K., Senthilvelan, M., Lakshmanan, M.: Chimeralike states in two distinct groups of identical populations of coupled Stuart-Landau oscillators. *Phys. Rev. E* **95**, 022208 (2017). <https://doi.org/10.1103/PhysRevE.95.022208>
22. Buscarino, A., Frasca, M., Gambuzza, L.V., Hövel, P.: Chimera states in time-varying complex networks. *Phys. Rev. E* **91**, 022817 (2015). <https://doi.org/10.1103/PhysRevE.91.022817>
23. Hagerstrom, A., Murphy, T.E., Roy, R., Hövel, P., Omelchenko, I., Schöll, E.: Experimental observation of chimeras in coupled-map lattices. *Nat. Phys.* **8**, 658 (2012). <https://doi.org/10.1038/nphys2372>
24. Rattenborg, N.C., Amlaner, C.J., Lima, S.L.: Behavioral, neurophysiological and evolutionary perspectives on unihemispheric sleep. *Neurosci. Biobehav. Rev.* **24**, 817–842 (2000). [https://doi.org/10.1016/S0149-7634\(00\)00039-7](https://doi.org/10.1016/S0149-7634(00)00039-7)
25. Filatrella, G., Neilson, A.H., Pedersen, N.F.: Analysis of a power grid using a Kuramoto-like model. *Eur. Phys. J. B* **61**(4), 485–491 (2008). <https://doi.org/10.1140/epjb/e2008-00098-8>

26. Shanahan, M.: Metastable chimera states in community-structured oscillator networks. *Chaos* **20**, 013108 (2010). <https://doi.org/10.1063/1.3305451>
27. Zakharova, A., Kapeller, M., Schöll, E.: Chimera death: symmetry breaking in dynamical networks. *Phys. Rev. Lett.* **112**, 154101 (2014). <https://doi.org/10.1103/PhysRevLett.112.154101>
28. Premalatha, K., Chandrasekar, V.K., Senthilvelan, M., Lakshmanan, M.: Different kinds of chimera death states in nonlocally coupled oscillators. *Phys. Rev. E* **93**, 052213 (2016). <https://doi.org/10.1103/PhysRevE.93.052213>
29. Premalatha, K., Chandrasekar, V.K., Senthilvelan, M., Lakshmanan, M.: Impact of symmetry breaking in networks of globally coupled oscillators. *Phys. Rev. E* **91**, 052915 (2015). <https://doi.org/10.1103/PhysRevE.91.052915>
30. Daido, H., Nakanishi, K.: Diffusion-induced inhomogeneity in globally coupled oscillators: swing-by mechanism. *Phys. Rev. Lett.* **96**, 054101 (2006). <https://doi.org/10.1103/PhysRevLett.96.054101>
31. Gopal, R., Chandrasekar, V.K., Venkatesan, A., Lakshmanan, M.: Observation and characterization of chimera states in coupled dynamical systems with nonlocal coupling. *Phys. Rev. E* **89**, 052914 (2014). <https://doi.org/10.1103/PhysRevE.89.052914>



Evolution of Similar Configurations in Graph Dynamical Systems

Joshua D. Priest¹, Madhav V. Marathe¹, S. S. Ravi^{1,2}(✉), Daniel J. Rosenkrantz^{1,2}, and Richard E. Stearns^{1,2}

¹ University of Virginia, Charlottesville, USA
{jdp8jb,marathe}@virginia.edu

² University at Albany – SUNY, Albany, USA

ssravi0@gmail.com, drosenkrantz@gmail.com, thestearns2@gmail.com

Abstract. We investigate questions related to the time evolution of discrete graph dynamical systems where each node has a state from $\{0,1\}$. The configuration of a system at any time instant is a Boolean vector that specifies the state of each node at that instant. We say that two configurations are similar if the Hamming distance between them is small. Also, a predecessor of a configuration B is a configuration A such that B can be reached in one step from A. We study problems related to the similarity of predecessor configurations from which two similar configurations can be reached in one time step. We address these problems both analytically and experimentally. Our analytical results point out that the level of similarity between predecessors of two similar configurations depends on the local functions of the dynamical system. Our experimental results, which consider random graphs as well as small world networks, rely on the fact that the problem of finding predecessors can be reduced to the Boolean Satisfiability problem (SAT).

1 Introduction

Discrete graph dynamical systems are generalizations of cellular automata (CA) [10,26]. They serve as a useful formal model in many contexts, including multi-agent systems, propagation of contagions in social networks and interaction phenomena in biological systems (see e.g., [1,17,25,27]). Here, we focus on one such class of graph dynamical systems, namely *synchronous* discrete dynamical systems (SyDSs). Informally, a SyDS¹ consists of an undirected graph² whose vertices represent entities and edges represent local interactions among entities. Each node v has a Boolean state and a local function f_v whose inputs are the current state of v and those of its neighbors; the output of f_v is the next state of v . The vector consisting of the state values of all the nodes at each time instant is referred to as the **configuration** of the system at that instant. In each time step, all nodes of a SyDS compute and update their states *synchronously*. Starting from a (given) initial configuration, the time evolution of a SyDS consists of a sequence of successive configurations, which is also called a **trajectory**.

¹ Formal definitions associated with SyDSs are presented in Sect. 2.

² Synchronous dynamical systems, where the underlying graph is directed, are called **Synchronous Boolean Networks** (see e.g., [12,13,19]).

In this paper, we examine questions related to the evolution of configurations that are *similar*. We measure the similarity between two configurations by their Hamming distance (i.e., the number of bit positions where the two configurations differ). Thus, two configurations are similar if the Hamming distance between them is small. It is known that certain dynamical systems may exhibit unpredictable behavior when the initial conditions are perturbed slightly [26]. A primary goal of our study is to obtain an understanding of when and how two similar configurations may arise from configurations that may be dissimilar. Such a study can be useful in understanding the sensitivity of a given dynamical system. As a concrete and simplified version of the general research question, we consider the following problem: given two similar configurations, how similar are their *predecessors* (i.e., configurations that just preceded the given configurations in the time evolution of the system)? A summary of our results is given below.

(a) Analytical Results. In Sect. 3, we show that SyDSs may exhibit extreme behaviors with respect to the evolution of configurations. For example, one of our results (Proposition 1) shows that there are SyDSs in which for any two configurations \mathbb{C}_1 and \mathbb{C}_2 which differ in h bits, there are respective predecessors \mathbb{C}'_1 and \mathbb{C}'_2 which also differ in exactly h bits. Further, we show (Proposition 2) that there are SyDSs where two very similar configurations (which differ in just one bit) have highly dissimilar predecessors (i.e., they differ in all the bits). In addition, we present examples of SyDSs (Corollary 1) in which highly dissimilar configurations have predecessors that differ in just one bit. We also show that computing similarity measures of the predecessors of two given configurations is, in general, computationally intractable. Further, we point out that the problem of computing a predecessor of a given configuration can be reduced to the Boolean Satisfiability problem (SAT).

(b) Experimental Results. Our experimental results (presented in Sect. 4) rely on the result that the problem of computing a predecessor of a given configuration can be reduced to SAT. While many public domain SAT solvers are available [22], we used Clasp [7] for our experiments. The reasons for this choice are explained in Sect. 4. Our experiments consider several classes of graphs (grids, Watts-Strogatz small world networks and Erdős-Rényi graphs). Since our analytical results indicate that non-monotone Boolean functions (e.g., exclusive OR) can cause extreme behaviors with respect to Hamming distance, we used threshold³ functions (which are monotone) in our experiments. For small networks, our results show the exact maximum, minimum and average Hamming distance values for several threshold values. For larger networks, since it is computationally expensive to find all the predecessors and compute the exact Hamming distance values, we generated up to 10^4 predecessors and computed the Hamming distance values using those predecessors. In general, the results discussed in Sect. 4 indicate that for small threshold values, as the Hamming distance between a pair of configurations is increased, the average Hamming distance between their predecessor sets increases linearly; for larger threshold values, the average Hamming distance between predecessor sets remains more or less stable. We also present results showing the number of clauses generated by the transformation of the predecessor problem into SAT

³ The class of threshold functions is defined in Sect. 2.

and the time used by two SAT solvers (namely, Clasp [7] and Glucose [8]) to solve the corresponding SAT instances.

Related Work. Computational problems associated with discrete dynamical systems have been addressed by many researchers. For example, Barrett et al. [3] and Rosenkrantz et al. [21] studied the reachability problem (i.e., given a SyDS \mathbb{S} and two configurations \mathbb{C}_1 and \mathbb{C}_2 , does \mathbb{S} starting from \mathbb{C}_1 reach \mathbb{C}_2 ?) for undirected graphs. The same problem for directed graphs has been studied in [5, 19]. Tosić [23, 24] presented results for counting the number of fixed points⁴ for systems with special forms of local functions. Kosub and Homan [15] presented dichotomy results that delineate computationally intractable and efficiently solvable versions of counting fixed points, based on the class of allowable local functions. The complexity of the predecessor existence problem for various classes of underlying graphs and local functions is investigated in [4]. A more general version of the predecessor existence problem, where the goal is to find t -step predecessors for values of $t \geq 2$, has been studied in [14, 16]. Problems similar to predecessor existence have also been considered in the context of cellular automata [6, 9]. Readers interested in the applications of graph dynamical systems are referred to [1, 17].

Note: For space reasons, proofs are not included; they can be found in [20].

2 Preliminaries

Synchronous Dynamical Systems and Local Functions. We follow the presentation in [4] for the basic definitions associated with discrete dynamical systems. Let \mathbb{B} denote the Boolean domain $\{0, 1\}$. A **Synchronous Dynamical System** (SyDS) \mathbb{S} over \mathbb{B} is specified as a pair $\mathbb{S} = (G, \mathbb{F})$, where (a) $G(V, E)$, an undirected graph with $|V| = n$, represents the underlying graph of the SyDS and (b) $\mathbb{F} = \{f_1, f_2, \dots, f_n\}$ is a collection of functions in the system, with f_i denoting the **local function** associated with node v_i , $1 \leq i \leq n$. Each node of G has a state value from \mathbb{B} . For any node v , we use $N[v]$ to denote the **closed neighborhood** of v , that is, the set consisting of v and all its neighbors. Each function f_i specifies the local interaction between node v_i and its neighbors in G . The inputs to function f_i are the state of the nodes in $N[v_i]$; function f_i maps each combination of inputs to a value in \mathbb{B} . This value becomes the next state of node v_i . It is assumed that each local function can be computed efficiently.

At any time τ , the **configuration** \mathbb{C} of a SyDS is the n -vector $(s_1^\tau, s_2^\tau, \dots, s_n^\tau)$, where $s_i^\tau \in \mathbb{B}$ is the state of node v_i at time τ ($1 \leq i \leq n$). Given a configuration \mathbb{C} , the state of a node v in \mathbb{C} is denoted by $\mathbb{C}(v)$. In a SyDS, all nodes compute and update their next state *synchronously*. Other update disciplines (e.g., sequential updates) have also been considered in the literature (e.g., [4, 17]). Suppose a given SyDS transitions in one step from a configuration \mathbb{C}' to a configuration \mathbb{C} . Then we say that \mathbb{C} is the **successor** of \mathbb{C}' , and \mathbb{C}' is a **predecessor** of \mathbb{C} . Since the SyDSs considered in this paper are deterministic, each configuration has a *unique* successor. However, a configuration may have zero or more predecessors. In the graph dynamical systems literature, configurations with no

⁴ A fixed point of a SyDS is a configuration which is its own successor.

predecessors are called **Garden of Eden** (GE) configurations [17]. Given a configuration \mathbb{C} , we use the notation $\sigma(\mathbb{C})$ to denote the successor of \mathbb{C} , and $\Pi(\mathbb{C})$ to denote the set of all predecessors of \mathbb{C} .

SyDSs have been considered in the literature under many classes of local functions (see e.g., [4, 14]). We now present an example of a SyDS where the local function at each node is a **threshold function**. For each integer $k \geq 0$, the k -**threshold function** has the value 1 iff at least k of its inputs are 1.

Example: The underlying graph of a SyDS shown in Fig. 1. The threshold value for each node is shown within parentheses. (Thus, the local function at b is the 2-threshold function while that at d is the 3-threshold function.) Suppose the initial configuration of the system is $(1, 1, 0, 0, 0)$; that is, a and b are in state 1 while c, d and e are in state 0. The reader can verify that starting from time 0, the system goes through the following sequence of configurations: $(1, 1, 0, 0, 0) \rightarrow (1, 1, 1, 0, 0) \rightarrow (1, 1, 1, 1, 0) \rightarrow (1, 1, 1, 1, 1)$. Once the system reaches the configuration $(1, 1, 1, 1, 1)$ at time step 3, no further state changes occur in the subsequent time steps; that is, the configuration $(1, 1, 1, 1, 1)$ is a **fixed point**.

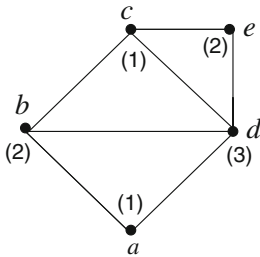


Fig. 1. An Example of a SyDS where each node has a threshold function. The threshold values are shown in parentheses.

The **phase space** $\mathbb{P}_{\mathbb{S}}$ of a SyDS \mathbb{S} is a directed graph defined as follows. There is a node in $\mathbb{P}_{\mathbb{S}}$ for each configuration of \mathbb{S} . There is a directed edge from a node representing configuration \mathbb{C}_1 to that representing configuration \mathbb{C}_2 if there is a one step transition of \mathbb{S} from \mathbb{C}_1 to \mathbb{C}_2 . For a SyDS with n nodes, the number of nodes in the phase space is 2^n ; thus, the size of phase space is *exponential* in the size of a SyDS. Each node in the phase space has an outdegree of 1 (since our SyDS model is deterministic). Also, in the phase space, each fixed point of a SyDS is a self-loop and each GE configuration is a node of indegree zero.

Hamming Distance and Similarity of Configurations. Given two configurations \mathbb{C}_1 and \mathbb{C}_2 of a SyDS over the domain $\{0, 1\}$, the **Hamming Distance** between \mathbb{C}_1 and \mathbb{C}_2 , denoted by $\mathbb{H}(\mathbb{C}_1, \mathbb{C}_2)$, is the number of positions in which they differ. For example, if $\mathbb{C}_1 = (1, 0, 0, 1)$ and $\mathbb{C}_2 = (0, 1, 0, 0)$, then $\mathbb{H}(\mathbb{C}_1, \mathbb{C}_2) = 3$. We say that two configurations \mathbb{C}_1 and \mathbb{C}_2 of a SyDS are h -**close** if $\mathbb{H}(\mathbb{C}_1, \mathbb{C}_2) = h$. Two configurations that are h -close for a small value of h can be thought of as ‘similar’ configurations. We note that in a SyDS with n nodes, the maximum Hamming distance between any pair of configurations \mathbb{C}_1 and \mathbb{C}_2 is n ; this occurs when \mathbb{C}_1 is the bitwise complement of \mathbb{C}_2 .

Similarity Measures for Sets of Configurations. Our focus is on studying the degree of similarity between predecessors of similar configurations. To do this, we define the following distance measures between two sets of nonempty configurations S_1 and S_2 .

(a) **Minimum Separation (MINSEP)**: This measure is defined as follows:

$$\text{MINSEP}(S_1, S_2) = \min\{\mathbb{H}(\mathbb{C}, \mathbb{C}') : \mathbb{C} \in S_1, \mathbb{C}' \in S_2\}.$$

(b) **Maximum Separation (MAXSEP)**: This measure, which is analogous to minimum separation, is defined as follows.

$$\text{MAXSEP}(S_1, S_2) = \max\{\mathbb{H}(\mathbb{C}, \mathbb{C}') : \mathbb{C} \in S_1, \mathbb{C}' \in S_2\}.$$

(c) **Average Separation (AVGSEP)**: This measure is defined as follows.

$$\text{AVGSEP}(S_1, S_2) = \frac{\sum_{\mathbb{C} \in S_1, \mathbb{C}' \in S_2} \mathbb{H}(\mathbb{C}, \mathbb{C}')}{|S_1| \times |S_2|}.$$

Among the above measures, a small value of MAXSEP provides the strongest guarantee of similarity. This is because if $\text{MAXSEP}(S_1, S_2) = \alpha$, and α is small, then the Hamming distance between any pair configurations \mathbb{C} and \mathbb{C}' , where $\mathbb{C} \in S_1$ and $\mathbb{C}' \in S_2$, is at most α ; in other words, each such configuration pair is α -close. For convenience, when at least one of the sets S_1 and S_2 is empty, we define the values of $\text{MINSEP}(S_1, S_2)$, $\text{MAXSEP}(S_1, S_2)$ and $\text{AVGSEP}(S_1, S_2)$ to be ∞ .

The following lemma points out two simple properties of predecessors in SyDSs.

Lemma 1. *Let \mathbb{S} be a SyDS. (i) Suppose \mathbb{C}_1 and \mathbb{C}_2 are two different configurations of \mathbb{S} . The sets $\Pi(\mathbb{C}_1)$ and $\Pi(\mathbb{C}_2)$ are disjoint. (ii) Suppose every configuration of \mathbb{S} has a predecessor. Then each configuration of \mathbb{S} has a unique predecessor.*

Proof: See [20].

Boolean Satisfiability Problem (SAT): Given an m -variable Boolean function F of in conjunctive normal form (CNF), the goal of the Satisfiability problem (SAT) problem is to determine whether there is an assignment of a Boolean values to each of the m variables so that the function F evaluates to true under the assignment. We will explain in Sect. 3 how the problem of finding predecessors of a given configuration can be reduced to an appropriate instance of SAT. Many public domain SAT solvers are currently available to obtain solutions to practical SAT instances [22]. Our experimental results in Sect. 4 were generated using SAT solvers.

3 Analytical Results

Overview. In this section, we first show that the problem of finding the predecessors of a given configuration of a SyDS can be expressed as an instance of SAT. This transformation forms the basis for the experimental results presented in Sect. 4. In addition, we present several analytical results regarding the similarities of predecessor sets of two configurations of a SyDS. Throughout this section, the reader should bear in mind that for any configuration \mathbb{C} , $\sigma(\mathbb{C})$ denotes the successor of \mathbb{C} and $\Pi(\mathbb{C})$ denotes the set of all predecessors of \mathbb{C} .

Reducing Predecessor Finding to SAT. We assume that the nodes of the underlying graph of the given SyDS are numbered 1 through n and that the local function at node

i is denoted by f_i , $1 \leq i \leq n$. For each node i , let N_i denote the **closed neighborhood** of node i (defined in Sect. 2) in the underlying graph; thus, the states of the nodes in N_i are the inputs to the local function f_i , $1 \leq i \leq n$.

Let $\mathbb{C} = (c_1, c_2, \dots, c_n)$ be the given configuration for which we need to find a predecessor (if one exists). Note that each c_i is a known 0 or 1 value, $1 \leq i \leq n$. We need to find a configuration $\mathbb{C}' = (x_1, x_2, \dots, x_n)$ such that \mathbb{C}' is a predecessor of \mathbb{C} (if one exists). This condition can be transformed into an instance of SAT as follows.

Consider node i of the SyDS. As mentioned earlier, let $N_i = \{i_1, i_2, \dots, i_r\}$ denote the closed neighborhood of node i , where $r = |N_i|$. Thus, the inputs to the local function f_i at node i are $x_{i_1}, x_{i_2}, \dots, x_{i_r}$. Since we want \mathbb{C}' to be a predecessor of \mathbb{C} , the condition to be satisfied at node i is the following:

$$c_i \Leftrightarrow f_i(x_{i_1}, x_{i_2}, \dots, x_{i_k}). \quad (1)$$

Since c_i is a known 0 or 1 value, the expression given in Eq. (1) can be simplified. If $c_i = 0$, the above expression simplifies to $\neg f_i(x_{i_1}, x_{i_2}, \dots, x_{i_k})$. Likewise, if $c_i = 1$, the above expression simplifies to $f_i(x_{i_1}, x_{i_2}, \dots, x_{i_k})$.

Using P_i to denote the subexpression given by Eq. (1) for node i , the condition to be satisfied for \mathbb{C}' to be a predecessor of \mathbb{C} is given by

$$P_1 \wedge P_2 \wedge \dots \wedge P_n. \quad (2)$$

As before, since each subexpression P_i can be expressed as an equivalent CNF, we can get a CNF formula with variables x_1, x_2, \dots, x_n from Eq. (2). Each solution to the resulting CNF formula (which can be obtained using a SAT solver) gives a predecessor of the given configuration \mathbb{C} . If there is no satisfying assignment to the CNF formula corresponding to the expression in Eq. (2), then \mathbb{C} has no predecessor; that is, \mathbb{C} is a Garden-of-Eden configuration. This SAT-based approach for finding predecessors will be incorporated into a software system called `net.science` that is being built in collaboration with several organizations [2].

Results on Similarities of Predecessor Sets. We now present our theoretical results regarding the similarity of predecessors of two configurations. Our first result points out that there are SyDSs where the Hamming distance between a pair of configurations is preserved when predecessors are considered.

Proposition 1. *Let G be an arbitrary graph. Then, there is a SyDS \mathbb{S} with underlying graph G , such that \mathbb{S} has the following properties: (i) every configuration has a predecessor, and (ii) for any pair of distinct configurations \mathbb{C}_1 and \mathbb{C}_2 , $\mathbb{H}(\sigma(\mathbb{C}_1), \sigma(\mathbb{C}_2)) = \mathbb{H}(\mathbb{C}_1, \mathbb{C}_2)$ and $\text{MAXSEP}(\Pi(\mathbb{C}_1), \Pi(\mathbb{C}_2)) = \text{MINSEP}(\Pi(\mathbb{C}_1), \Pi(\mathbb{C}_2)) = \text{AVGSEP}(\Pi(\mathbb{C}_1), \Pi(\mathbb{C}_2)) = \mathbb{H}(\mathbb{C}_1, \mathbb{C}_2)$.*

Proof: See [20].

Our next result shows that there are SyDSs for which there are two distinct configurations that are 1-close, but their predecessors are highly dissimilar; that is, they have the maximum possible Hamming distance.

Proposition 2. *Let G be an arbitrary connected graph, and let n be the number of nodes in G . Then, there is a SyDS \mathbb{S} with underlying graph G , such that \mathbb{S} has the following properties: (i) every configuration has a predecessor and (ii) for every configuration \mathbb{C}_1 , there is a configuration \mathbb{C}_2 such that $\mathbb{H}(\mathbb{C}_1, \mathbb{C}_2) = 1$ and $\text{MAXSEP}(\Pi(\mathbb{C}_1), \Pi(\mathbb{C}_2)) = \text{MINSEP}(\Pi(\mathbb{C}_1), \Pi(\mathbb{C}_2)) = \text{AVGSEP}(\Pi(\mathbb{C}_1), \Pi(\mathbb{C}_2)) = n$.*

Proof: See [20].

We now show the existence of SyDSs in which there are pairs of configurations which have the maximum level of dissimilarity but their predecessors are 1-close.

Proposition 3. *Let G be an arbitrary graph, and let Δ be the maximum node degree of G . Then, there is a SyDS \mathbb{S} with underlying graph G , such that \mathbb{S} has the following properties: (i) every configuration has a predecessor and (ii) for every configuration \mathbb{C}_1 , there is a configuration \mathbb{C}_2 such that $\mathbb{H}(\mathbb{C}_1, \mathbb{C}_2) = \Delta + 1$ and $\text{MAXSEP}(\Pi(\mathbb{C}_1), \Pi(\mathbb{C}_2)) = \text{MINSEP}(\Pi(\mathbb{C}_1), \Pi(\mathbb{C}_2)) = \text{AVGSEP}(\Pi(\mathbb{C}_1), \Pi(\mathbb{C}_2)) = 1$.*

Proof: See [20].

The following corollary is a direct consequence of Proposition 3 by taking the underlying graph of the SyDS to be the star graph on n nodes.

Corollary 1. *For any integer $n \geq 2$, there is a SyDS \mathbb{S} with n nodes satisfying the following properties: (i) there is a pair of configurations \mathbb{C}_1 and \mathbb{C}_2 with $\mathbb{H}(\mathbb{C}_1, \mathbb{C}_2) = n$ and $\text{MAXSEP}(\Pi(\mathbb{C}_1), \Pi(\mathbb{C}_2)) = 1$.*

We now present a result that establishes the computational complexity of computing distance measures for predecessor configurations. The decision problem, which we call **Minimum Predecessor Separation** (MPS), is the following: given a SyDS \mathbb{S} , two configurations \mathbb{C}_1 and \mathbb{C}_2 , and a positive integer q , is $\text{MINSEP}(\Pi(\mathbb{C}_1), \Pi(\mathbb{C}_2)) \leq q$? Using the known result that the **Predecessor Existence** problem (i.e., given a SyDS \mathbb{S} and a configuration \mathbb{C} , does \mathbb{C} have a predecessor?) is **NP**-complete [4], it can be shown that MPS is also **NP**-complete. This result is stated below.

Proposition 4. *The MPS problem is **NP**-complete.*

Proof: See [20].

Our proof of Proposition 4 relies on the fact that it **NP**-hard to decide whether a configuration \mathbb{C} has a predecessor. We now present a stronger **NP**-completeness result. We show that the MPS problem is **NP**-complete even when we are given predecessors of \mathbb{C}_1 and \mathbb{C}_2 . We call the decision problem when this extra information is given **Minimum Predecessor Separation Given Predecessors** (MPSGP). Note that since the predecessors of \mathbb{C}_1 and \mathbb{C}_2 are specified in a given MPSGP problem instance, it is unnecessary to explicitly specify \mathbb{C}_1 and \mathbb{C}_2 . Thus, we formalize the MPSGP problem as follows: given a SyDS \mathbb{S} , and two configurations \mathbb{C}'_1 and \mathbb{C}'_2 , is $\text{MINSEP}(\Pi(\sigma(\mathbb{C}'_1)), \Pi(\sigma(\mathbb{C}'_2))) < \mathbb{H}(\mathbb{C}'_1, \mathbb{C}'_2)$? Our next result points out the **NP**-hardness of this problem.

Theorem 1. *The MPSGP problem is **NP**-complete.*

Proof: See [20].

4 Experimental Results

Overview. The analytical results presented in Sect. 3 show that in general, SyDSs may exhibit extreme behaviors with respect to evolution of configurations. So, in the experimental phase, our goal was to understand the behavior for restricted classes of graphs and local functions. We generated SyDSs whose underlying graphs are from special classes of graphs and whose local functions are from restricted classes of Boolean functions. We generated pairs of configurations that are h -close for small values of h and examined the range of Hamming distances for their sets of predecessors. We used the transformation from the predecessor problem to SAT discussed in Sect. 3.

SyDS Construction. We investigated several types of underlying graph structures including Erdős–Rényi models, lattice/grid graphs, and Watts–Strogatz small-world networks [18]. All graphs were created using the NetworkX library [11]. The Erdős–Rényi graphs were constructed such that the estimated mean degree of the graph was 16. Grid graphs were constructed such that each node connected to exactly four other nodes. Nodes in the Watts–Strogatz small world networks were initially wired to their eight nearest neighbors; then each edge had a 50% chance to be rewired to a random node in the graph.

To examine the similarity of configurations, we considered several local functions. All SyDSs constructed and tested were uniform SyDSs⁵ with threshold functions ranging from threshold 1 (equivalent to Boolean OR) to threshold 4. We chose threshold functions as they are monotone Boolean functions. As shown in Sect. 3, SyDSs with similar configurations and non-monotone local functions (such as exclusive OR) can have predecessors with very high variability in their Hamming distances. With threshold functions, we expected the Hamming distances of the predecessors of similar configurations to show less extreme variance.

Procedure for Generating Configurations and Their Predecessors. We implemented the transformation from the predecessor problem to SAT in Python. We limited the number of predecessors generated for each configuration for the following reasons. In order to compute the minimum, average, and maximum Hamming distances between two sets S_1 and S_2 of predecessors, each predecessor in S_1 must be compared with each predecessor from S_2 . For example, with just 10^4 predecessors for each configuration, the number of such comparisons is 10^8 . In addition to time used for such a computation, attempting to exhaustively find and record every predecessor for larger graph sizes could generate several terabytes of data.

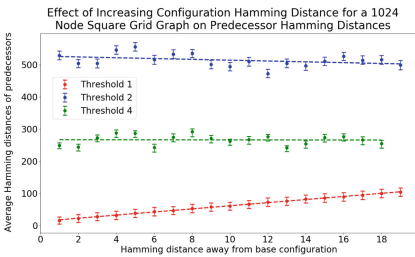
We defined our “base” configuration as the one with all node states set to 1. To generate a configuration with Hamming distance h from the base configuration, the states of h random nodes were changed from 1 to 0. In total, 20 configurations with different Hamming distances were generated. We generated up to 10^4 solutions for each predecessor problem. We computed the necessary Hamming distance values between the set of predecessors for the base configuration and the sets of predecessors of the 20 configurations derived from the base configuration. Our results provide an indication of the minimum and maximum Hamming distances. In the plots shown in this section, the

⁵ A **uniform** SyDS is one in which all the nodes have the same local function.

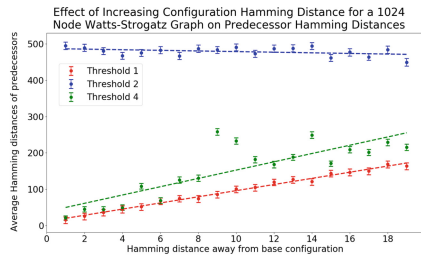
mean Hamming distance between the predecessors of the base configuration and those of the 20 derived configurations are shown, with error bars representing the minimum and maximum Hamming distances of the solution sets. For each threshold value, we fit a linear trend line to the results.

Table 1. Table showing minimum, maximum and average Hamming distance values for grids and Watts-Strogatz small world networks with 16 nodes

Threshold	Hamming distance from base Configuration	Square Grid			Watts-Strogatz Network		
		Predecessors' Hamming distance			Predecessors' Hamming distance		
		Minimum	Average	Maximum	Minimum	Average	Maximum
2	2	2	8.000	14	1	8.248	16
	4	2	8.376	16	1	8.304	16
	6	2	8.602	16	2	8.384	16
	14	5	9.605	16	3	8.490	16
3	2	2	6.905	11	1	8.250	16
	4	2	6.905	11	2	8.537	16
	6	2	7.502	13	1	8.473	16
	12	5	9.095	14	2	8.799	16
	14	5	9.540	15	4	8.883	16
4	2	1	3.789	5	1	7.872	15
	4	2	4.491	6	1	7.964	14
	10	4	6.421	8	3	8.486	15
	12	4	7.013	10	4	9.041	16
	14	4	8.191	12	5	9.163	15



(a) Graph showing average Hamming distance values for a 1024 node square grid network



(b) Graph showing average Hamming distance values for a 1024 node Watts-Strogatz network

Fig. 2. Average Hamming distance values for grid and Watts-Strogatz networks

Hamming Distance Results for Small Networks. Table 1 shows the minimum, average, and maximum Hamming distances for 16 node grids and Watts-Strogatz networks. For these small networks, we were able to generate all predecessors for each configuration. The table shows the results for the configurations for which both the grid and the Watts-Strogatz graph had predecessors. For both classes of graphs and all threshold values, the minimum and average predecessor Hamming distance show a roughly monotonic non-decreasing trend with increase in the Hamming distance of a configuration from the base configuration. The maximum Hamming distance also increased

monotonically for the grid graphs; however, for the Watts-Strogatz networks started at the highest value (16) and stayed very close to that value.

Hamming Distance Results for Large Networks. Our results for the 1024 node square grid network and the 1024 node Watts-Strogatz small world network are shown in Figs. 2a and 2b respectively. The average Hamming distances of predecessors for these two graphs show similar trends. For both networks, the Hamming distance values for threshold 1 were lower compared to the other threshold values; moreover, the average Hamming distance increased linearly with increase in the Hamming distance of a configuration from the base configuration. Threshold 2 showed the highest values of average Hamming distances for both networks; further, the average Hamming distance also showed a stable trend as configuration Hamming distance was increased. For the square grid graph, threshold 4 also showed a stable trend. In contrast, threshold 4 results for the Watts-Strogatz graph show a linearly increasing trend similar to Threshold 1. For both networks, the range of minimum and maximum Hamming distances was within 50 units of the average.

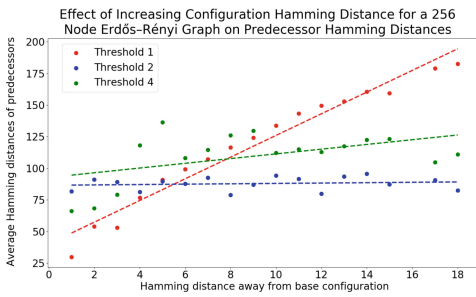


Fig. 3. Graph showing average Hamming distance values for a 256 node Erdős-Rényi network

Average Hamming distance values for an Erdős-Rényi graph with 256 nodes are shown in Fig. 3. There, the minimum and maximum Hamming distances in each set were within 30 units of the average and are not shown in Fig. 3 to avoid clutter. The average Hamming distance values for Threshold 1 once again show a linearly increasing trend with increase in the Hamming distance from the base configuration. Threshold 4 also shows a linearly increasing trend but with a slope smaller than that of threshold 1. The values for Threshold 2 show a more or less stable trend.

Number of Clauses Generated and SAT Solver Runtime. We conducted tests to compare the performance of the two most recently updated SAT solvers, namely Clasp [7] and Glucose [8]. For these experiments, graphs were generated in the same manner as previously mentioned except that Erdős-Rényi graphs for this experiment were constructed to have an average degree of 4. Assuming that each local function is the 1-threshold function, we computed the number of clauses generated for each predecessor problem with a uniform threshold of 1 on a sample of 20 predecessor problems and measured the average CPU time⁶ it took each SAT solver to produce one solution. The results are shown in Table 2.

There was no significant difference between the Glucose and Clasp SAT solvers in terms of CPU time taken to obtain a single solution to a SAT problem. The only notable exception is that for the larger Watts-Strogatz graph, Clasp was faster than Glucose

⁶ Experiments were run on a single core of a 2.80 GHz Intel Core i5-8400 CPU and with 16 GB of RAM.

Table 2. Table showing the number of clauses in the SAT instance generated from a predecessor problem and the CPU time to generate a solution for several networks

Network type	$2^{16} = 65,536$ Nodes			$2^{18} = 262,144$ Nodes		
	Number of clauses	Clasp time (seconds)	Glucose time (seconds)	Number of clauses	Clasp time (seconds)	Glucose time (seconds)
Square Grid	65806	0.059	0.054	262414	0.213	0.201
Watts-Strogatz	77614	0.552	0.772	299310	8.418	11.219
Erdős-Rényi	65686	0.096	0.129	262800	0.882	0.863

(8.418 s vs 11.219 s). The larger amount of time used for this graph could potentially be due to the larger average degree. Clasp was eventually chosen for our experiments because it can generate all the solutions for a given SAT instance.

5 Summary and Future Research Directions

We presented analytical and experimental results regarding the time evolution of similar configurations. We demonstrated the use of SAT solvers in studying these questions. There are several directions for future work. We considered one method of generating similar pairs of configurations starting from a base configuration. One may investigate other ways of generating similar configurations. Also, instead of considering one step predecessors, one may consider similarity issues for t -step predecessors for $t \geq 2$. Such generalized predecessor problems can also be reduced to SAT. Further, instead of Hamming distance, one may consider other measures of similarity between configurations; for example, two configurations may be considered similar if they have the same number of 1's.

Acknowledgments. We thank the referees for their comments. This work is partially supported by NSF Grants ACI-1443054 (DIBBS), IIS-1633028 (BIG DATA), CMMI-1745207 (EAGER), OAC-1916805 (CINES), CCF-1918656 (Expeditions) and IIS-1908530.

References

1. Adiga, A., Kuhlman, C.J., Marathe, M.V., Mortveit, H.S., Ravi, S.S., Vullikanti, A.: Graphical dynamical systems and their applications to bio-social systems. *Springer Int. J. Adv. Eng. Sci. Appl. Math.* **11**(2), 153–171 (2019)
2. Ahmed, N.K., Alo, R.A., Amelink, C.T., et al.: net.science: a cyberinfrastructure for sustained innovation in network science and engineering. In: *Gateway* (2020)
3. Barrett, C.L., Hunt III, H.B., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E.: Complexity of reachability problems for finite discrete dynamical systems. *J. Comput. Syst. Sci.* **72**(8), 1317–1345 (2006)
4. Barrett, C., Hunt III, H.B., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E., Thakur, M.: Predecessor existence problems for finite discrete dynamical systems. *Theoret. Comput. Sci.* **386**(1), 3–37 (2007)
5. Chistikov, D., Lisowski, G., Paterson, M., Turrini, P.: Convergence of opinion diffusion is PSPACE-complete. *CoRR abs/1912.09864* (2019). <http://arxiv.org/abs/1912.09864>
6. Durand, B.: A random NP-complete problem for inversion of 2D cellular automata. *Theoret. Comput. Sci.* **148**(1), 19–32 (1995)

7. Gebser, M., Kaufmann, B., Neumann, A., Schaub, T.: Clasp: a conflict-driven answer set solver. In: Baral, C., Brewka, G., Schlipf, J. (eds.) *Logic Programming and Nonmonotonic Reasoning*, pp. 260–265. Springer, Heidelberg (2007)
8. The Glucose SAT solver (2016). <https://www.labri.fr/perso/lsimon/glucoose/>
9. Green, F.: NP-complete problems in Cellular Automata. *Complex Syst.* **1**(3), 453–474 (1987)
10. Gutowitz, H.: *Cellular Automata: Theory and Experiment*. North Holland (1989)
11. Hagberg, A., Schult, D., Swart, P.: NetworkX reference (2020). https://networkx.github.io/documentation/latest/_downloads/networkx_reference.pdf
12. Kauffman, S., Peterson, C., Samuelsson, B., Troein, C.: Random Boolean network models and the yeast transcriptional network. *Proc. Natl. Acad. Sci. (PNAS)* **100**(25), 14796–14799 (2003)
13. Kauffman, S., Peterson, C., Samuelsson, B., Troein, C.: Genetic networks with canalizing Boolean rules are always stable. *Proc. Natl. Acad. Sci. (PNAS)* **101**(49), 17102–17107 (2004)
14. Kawachi, A., Ogihara, M., Uchizawa, K.: Generalized predecessor existence problems for Boolean finite dynamical systems. In: *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017)*, pp. 8:1–8:13 (2017)
15. Kosub, S., Homan, C.M.: Dichotomy results for fixed point counting in Boolean dynamical systems. In: *Proceedings of the 10th Italian Conference on Theoretical Computer Science*, pp. 163–174 (2007)
16. Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E.: Computational aspects of fault location and resilience problems for interdependent infrastructure networks. In: *International Conference on Complex Networks and their Applications*, pp. 879–890. Springer, Heidelberg (2018)
17. Mortveit, H., Reidys, C.: *An Introduction to Sequential Dynamical Systems*. Springer, New York (2007)
18. Newman, M., Barabási, A.L., Watts, D.J.: *The Structure and Dynamics of Networks*. Princeton University Press, Princeton (2006)
19. Ogihara, M., Uchizawa, K.: Computational complexity studies of synchronous Boolean finite dynamical systems on directed graphs. *Inf. Comput.* **256**, 226–236 (2017)
20. Priest, J.D., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E.: Evolution of similar configurations in graph dynamical systems. Technical Report for 2020, Network Systems Science and Advanced Computing (NSSAC) Division, Biocomplexity Institute and Initiative, University of Virginia, Charlottesville, VA, USA. <https://drive.google.com/file/d/1Bc2idtlFnk7uidLnDEi6U3iggET0O0dh/view?usp=sharing>
21. Rosenkrantz, D.J., Marathe, M.V., Ravi, S.S., Stearns, R.E.: Testing phase space properties of synchronous dynamical systems with nested canalizing local functions. In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, 10–15 July 2018*, pp. 1585–1594 (2018)
22. Information regarding SAT solvers (2018). <http://www.satlive.org>
23. Tasic, P.T.: On the complexity of enumerating possible dynamics of sparsely connected Boolean network automata with simple update rules. In: *Automata 2010 - 16th International Workshop on CA and DCS*, pp. 125–144 (2010)
24. Tasic, P.T.: Phase transitions in possible dynamics of cellular and graph automata models of sparsely interconnected multi-agent systems. In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, 8-12 May 2017*, pp. 474–483 (2017)
25. Valente, T.W.: Social network thresholds in the diffusion of innovations. *Soc. Netw.* **18**, 69–89 (1996)
26. Wolfram, S.: *Theory and Applications of Cellular Automata*. World Scientific (1987)
27. Wooldridge, M.: *An Introduction to Multi-Agent Systems*. Wiley, West Sussex (2002)



Congestion Due to Random Walk Routing

Onuttom Narayan¹, Iraj Saniee²(✉), and Vladimir Marbukh³

¹ Department of Physics, University of California, 1156 High Street,
Santa Cruz, CA 95064, USA
onarayan@ucsc.edu

² Mathematics and Algorithms Research, Bell Labs, Nokia, 600 Mountain Avenue,
Murray Hill, NJ 07974, USA
iis@research.bell-labs.com

³ Applied and Computational Mathematics, National Institute of Standards
and Technology, 100 Bureau Drive, Gaithersburg, MD 20899, USA
vladimir.marbukh@nist.gov

Abstract. In this paper we derive an analytical expression for the mean load at each node of an arbitrary undirected graph for the uniform multicommodity flow problem under random walk routing. We show the mean load is linearly dependent on the nodal degree with a common multiplier equal to the sum of the inverses of the non-zero eigenvalues of the graph Laplacian. Even though some aspects of the mean load value, such as linear dependence on the nodal degree, are intuitive and may be derived from the equilibrium distribution of the random walk on the undirected graph, the exact expression for the mean load in terms of the full spectrum of the graph has not been known before. Using the explicit expression for the mean load, we give asymptotic estimates for the load on a variety of graphs whose spectral density are well known. We conclude with numerical computation of the mean load for other well-known graphs without known spectral densities.

Keywords: Multicommodity flow · Network congestion · Steady state · Laplacian of a graph · Spectrum of a graph · Random walk

1 Introduction

The study of network capacity, sometimes referred to as load or congestion, is over half a century old, and goes back to the pioneering work of Ford and Fulkerson [1] and Shannon [2] for the single commodity and to early attempts [3–7] for the multicommodity flow solutions of the problem. This rather large literature provides a characterization of the load or, more specifically, the minimal capacity required, in terms of sum of link capacities needed based on cut values, which in case of the single commodity model are both necessary and sufficient and for the multicommodity case generally provide necessary conditions.

Single commodity or multicommodity network flow models in communication, transportation and numerous other settings typically assume shortest path

routing. There are natural settings in which alternative routing not involving shortest paths may be required. For example, it may happen that longer routes are used for load balancing or, in the case of capacitated networks, to avoid network expansion [8,9]. Or the inverse problem may be posed: to determine weights so that shortest path routes determined from these weights result in smallest load across the network [10]. Given the universality of the network flow model, there are a vast number of applications of the model, and the list is too large to enumerate here.

There are few analytical results concerning the multicommodity flow problem with shortest path routing, in the sense of having a closed form solution as a function of a small number of parameters characterizing the network and the commodities. These include characterization of the maximal load for hyperbolic graphs [11,12]. In this setting, for a network of N nodes one assumes 1 unit of (directed) flow between all $N(N - 1)$ node pairs, and then asks how the load scales due to shortest path routing as a function of N . This measure is sometimes referred to as the *betweenness centrality*, see [13].

In this paper, we study the near opposite of shortest path routing: when flows are routed in a uniformly random manner, each flow starting from its source and moving at each step randomly to a neighboring node and only stopping when the destination of the flow is reached. More specifically, we consider the case when one unit of traffic, or a single packet, is injected into the network at every time step at each node i for each possible destination node $j \neq i$. Thus there are $N(N - 1)$ units of traffic (or packets) injected into the N -node network at every time step. The network is assumed to be connected, i.e. have a single component. We first demonstrate that a steady-state distribution is achieved and then derive an expression for the expected flow, or the average number of packets passing through each node, in terms of the eigenvalues of the graph Laplacian. To illustrate the results more concretely, we estimate the largest mean loads for a few networks whose distribution of Laplacian eigenvalues are known. We note that similar but not identical measures to the expected load at each node have been investigated numerically in the context of node ranking, see [14].

2 General Results

2.1 Time Evolution Equations

As described in the previous section, we consider an undirected connected graph $G(N, E)$ with N nodes, in which packets of traffic are injected at various nodes in a deterministic manner and move towards specified destinations. The dynamics are discrete time, i.e. packets of traffic move from node to node at time $t = 0, 1, 2, 3, \dots$. At each time step, exactly $N - 1$ packets of unit size are injected into the graph at each node k , with one packet heading towards each other node in the graph $l \neq k$. Thus there are precisely $N(N - 1)$ packets injected into the graph at each time step. Any packet that is present at node i at time step t and whose destination is not i moves to one of the nodes adjacent to i at time $t + 1$. For this, one of the d_i nodes adjacent to i is chosen randomly, with probability

equal to $1/d_i$. However, any packet of traffic that is at its destination at time t is removed from the network, and is no longer present at time $t + 1$. Note that a packet that returns to its source as it moves around randomly continues as it would from any other node. The congestion or load at any node at any time step is a random variable equal to the number of packets that are being processed at that node. We are interested in the expected value of the number of packets at each node. We expect that in steady state, if and when it exists, packets are (injected and) removed from each node at the same rate, i.e. $N - 1$ packets per time step. We seek to find the steady state load, i.e. the average number of packets, at all the nodes of the network.

As a byproduct, we obtain the average time τ (or number of steps in its path) that a packet takes to go from a randomly chosen source node to a randomly chosen destination. A packet that hops from source to destination in t steps is in the network for t time steps. (We have assigned one time step each to the source and destination nodes.) The average number of packets at each node, summed over all the nodes in the graph, is therefore the product of the total injection rate $N(N - 1)$ and τ .

Remark 1. We shall use N to represent both the set of nodes in the graph as well as their count $|N|$ without danger of confusion. Also, we write $k \sim j$ to mean that node k is a neighbor of node j , i.e., i and j are adjacent, and $k \not\sim j$ when they are not; and refer to the adjacency matrix (A_{ij}) the Laplacian (L_{ij}) and the normalized Laplacian (\mathcal{L}_{ij}) (for $0 \leq i, j \leq N$) of the undirected graph $G(N, E)$, with their standard definitions:

$$A_{ij} = \begin{cases} 0, & i = j \\ 1, & i \sim j \\ 0, & i \not\sim j \end{cases}, \quad L_{ij} = \begin{cases} d_i, & i = j \\ -1, & i \sim j \\ 0, & i \not\sim j \end{cases}, \quad \mathcal{L}_{ij} = \begin{cases} 1, & i = j \\ -(d_i d_j)^{-\frac{1}{2}}, & i \sim j \\ 0, & i \not\sim j \end{cases} \tag{1}$$

Theorem 1. *For a connected graph $G(N, E)$ with deterministic injection rate of one packet at each node destined for each other node, where each packet is routed uniformly randomly from its current node to its neighbors until it reaches its destination, there exists a unique steady state number of packets at each node.*

Proof. We first consider the case of the traffic flowing from a single source node k to a single destination node l . Let $X_i^{kl}(t)$ be the random variable representing the number of packets at node i at time t and $Z_{ji}^{kl}(t + 1)$ be the random variable representing the number of packets sent out of node j , a neighbor of i , to i at time t . This assumes tacitly that an outgoing packet from node j that leaves j at time t reaches a neighboring node i at time $t + 1$; an incoming packet to node i from node j that reaches i at time t must leave node j at time $t - 1$. Then the boundary condition Eq. (2), and the no-escape condition from destination l Eq. (3), both hold:

$$X_i^{kl}(0) = \delta_{ik}, \quad 0 \leq i \leq N \tag{2}$$

$$Z_{li}^{kl}(t) = 0, \quad \forall t \geq 0. \tag{3}$$

Flow balance for outgoing packets implies that for all neighbors i of a node $j \neq l$,

$$X_j^{kl}(t) = \sum_{i \sim j \neq l} Z_{ji}^{kl}(t+1), \quad 1 \leq i, j \leq N, 0 \leq t. \tag{4}$$

which simply states that packets at node j at time t move out to its incident links at time $t + 1$. These same packets arrive at time $t + 1$ at adjacent nodes

$$X_i^{kl}(t+1) = \delta_{ik} + \sum_{l \neq j \sim i} Z_{ji}^{kl}(t+1), \quad 1 \leq i, j \leq N, 0 \leq t, \tag{5}$$

Notice that the first term on the right hand side of Eq. (5) accounts for the fact that one packet is injected at node k for destination l at each time step. The second term represents the packets that move to node i at time $t + 1$ from adjacent nodes at time t . The sum in this term excludes the node l because any packet that was at the node l (the destination) at time t is removed from the network and is no longer present at time $t + 1$.

Further, our assumption of uniformly random routing of packets from each node to its neighbors implies that for any neighbor i of a node $j \neq l$,

$$\mathbb{P}\{Z_{ji}^{kl}(t+1) = z\} = \binom{X_j(t)}{z} \left(\frac{1}{d_j}\right)^z \left(1 - \frac{1}{d_j}\right)^{X_j(t)-z}, \quad 0 \leq z \leq X_j(t), 0 \leq t. \tag{6}$$

Taking ensemble expectation of Eqs. (6) and (5) and using the standard expression for the mean of the binomial distribution for Eq. (6), we get that for all $0 \leq i, j \leq N$

$$\mathbb{E}[Z_{ji}^{kl}(t+1)] = \frac{1}{d_j} \mathbb{E}[X_j^{kl}(t)], \quad l \neq j \sim i \tag{7}$$

$$\mathbb{E}[X_i^{kl}(t+1)] = \delta_{ki} + \sum_{l \neq j \sim i} \mathbb{E}[Z_{ji}^{kl}(t+1)] \tag{8}$$

and substituting from Eq. (7) into (8), we get

$$\mathbb{E}[X_i^{kl}(t+1)] = \delta_{ik} + \sum_{l \neq j \sim i} \frac{\mathbb{E}[X_j^{kl}(t)]}{d_j} \tag{9}$$

or alternatively stated in terms of the adjacency matrix A_{ij} of the graph,

$$\mathbb{E}[X_i^{kl}(t+1)] = \delta_{ik} + \sum_{j \neq l} A_{ij} \frac{\mathbb{E}[X_j^{kl}(t)]}{d_j}. \tag{10}$$

Now define $p_i^{kl}(t) = (1 - \delta_{il})\mathbb{E}[X_j^{kl}(t)]$. In other words, $p_i^{kl}(t) = \mathbb{E}[X_j^{kl}(t)]$ except for the destination node, $i = l$, where $p_l^{kl} = 0$. The sum in Eq. (10) can now be unrestricted for $i \neq l$. The rate equation for the p_i 's is

$$p_i^{kl}(t+1) = \delta_{ik} + \sum_j A_{ij} \frac{p_j^{kl}(t)}{d_j} \tag{11}$$

for $i \neq l$, with the boundary condition $p_i^{kl}(t + 1) = 0$. The restricted sum in Eq. (9) has been replaced by an unrestricted sum in Eq. (11), but the l 'th node is now outside the domain of the equation. The boundary condition is an example of a Dirichlet boundary condition, where a function is defined in a region and is specified to be zero on the boundary of the region; in this case, the boundary is the node l and the region is all the other nodes in the graph.

We now show that, under the time evolution of Eq. (11), the function $p_i^{kl}(t)$ reaches a t -independent unique steady state. Let $p_i^{kl(1)}(0)$ and $p_i^{kl(2)}(0)$ be two initial configurations at $t = 0$, that evolve according to Eq. (11). Define $q_i^{kl}(t)$ to be equal to $[p_i^{kl(1)}(t) - p_i^{kl(2)}(t)]/\sqrt{d_i}$. Then q^{kl} satisfies

$$q_i^{kl}(t + 1) = \sum_j A_{ij} \frac{q_j^{kl}(t)}{\sqrt{d_j d_i}} \tag{12}$$

with the Dirichlet boundary condition at $i = l$. This is equivalent to $q_i^{kl}(t + 1) = \sum_j (\delta_{ij} - \mathcal{L}_{ij}) q_j^{kl}(t)$, where \mathcal{L} is the normalized Laplacian. Since \mathcal{L} is a real symmetric matrix, it has a complete set of eigenfunctions. The eigenvalues are all in the interval $0 \leq \lambda \leq 2$, with an eigenvalue at $\lambda = 0$ iff one can construct a function f on the graph for which $f_i = f_j$ for all nodes (i, j) , and an eigenvalue at $\lambda = 2$ iff one can construct f such that $f_i = -f_j$ whenever $j \sim i$ [15]. With Dirichlet boundary conditions, since $f = 0$ on the boundary nodes, both of these are impossible, and therefore $0 < \lambda < 2$. Thus the operator $I - \mathcal{L}$ (with Dirichlet boundary conditions) is a contraction. Therefore $q^{kl}(t \rightarrow \infty) \rightarrow 0$, and as $t \rightarrow \infty$ all initial configurations tend to the same t -independent steady state configuration. \square

2.2 Steady State Solution

In this section, we solve the fixed point of the time evolution Eq. (11) with Dirichlet boundary condition as introduced in the proof of Theorem (1). As before, $\{\lambda_\alpha, \alpha < N\}$ represent the eigenvalues of the graph Laplacian.

Theorem 2. *For a connected graph $G(N, E)$ with deterministic injection rate of $(N - 1)$ packets at each node destined for every other node, where each packet is routed uniformly randomly from its current node to its neighbors until it reaches its destination, the unique steady state number of packets at each node j is given by Λ_j where*

$$\Lambda_j = (N - 1) + Nd_j \sum_{\alpha \neq 0} \frac{1}{\lambda_\alpha}. \tag{13}$$

Proof. In steady state, we know that the load flowing into the node l at any time step must be equal to the load injected into the node k , i.e. unity. Therefore $\sum A_{lj} p_j^{kl} / d_j = 1$, and we can extend Eq. (11) as

$$p_i^{kl} = \delta_{ik} - \delta_{il} + \sum_j A_{ij} \frac{p_j^{kl}}{d_j} \tag{14}$$

for all i , with the additional condition $p_l^{kl} = 0$. It may seem that we have gained nothing by restricting our analysis to the steady state configuration, since we still have to impose Dirichlet boundary conditions at the l 'th node. However, as we shall see immediately, the solution to Eq. (14) can easily be found in terms of the eigenvectors of the Laplacian without the Dirichlet boundary condition, i.e. independent of k and l .

In order to convert Eq. (14) to a Hermitean eigenvalue problem, we define $p_j^{kl} = d_j r_j^{kl}$ and $L_{ij} = d_j \delta_{ij} - A_{ij}$. Then

$$\sum_j L_{ij} r_j^{kl} = \delta_{ik} - \delta_{il} \tag{15}$$

with $r_l^{kl} = 0$. Here (L_{ij}) is the Laplacian for the graph. Since (L_{ij}) is a real symmetric matrix, it has a complete set of real eigenvalues λ_α and real orthonormal eigenvectors ξ^α for $\alpha = 0, 1, 2, \dots, N - 1$. Using the standard properties of the Laplacian, all the eigenvalues are non-negative, and since the graph has been assumed to have one component, there is only one zero eigenvalue λ_0 with eigenvector $\xi^0 = (1, 1, 1, \dots, 1)/\sqrt{N}$. The denominator ensures that the normalization condition $\sum_i \xi_i^0 \xi_i^0 = 1$ is satisfied.

We define

$$\pi_{kl}^\alpha = \sum_i \xi_i^\alpha (\delta_{ik} - \delta_{il}) = \xi_k^\alpha - \xi_l^\alpha \tag{16}$$

which is the projection of the right hand side of Eq. (15) on to the α 'th eigenvector. Note that $\pi_{kl}^0 = 0$. With this definition,

$$r_j^{kl} = \sum_{\alpha=1}^{N-1} \frac{\pi_{kl}^\alpha}{\lambda_\alpha} \xi_j^\alpha + c^{kl} \xi_j^0, \tag{17}$$

where c^{kl} has to be chosen to make r_l^{kl} equal to zero. Since ξ_j^0 is independent of j , the condition $r_l^{kl} = 0$ yields

$$r_j^{kl} = \sum_{\alpha=1}^{N-1} \frac{\xi_k^\alpha - \xi_l^\alpha}{\lambda_\alpha} \xi_j^\alpha - \sum_{\alpha=1}^{N-1} \frac{1}{\lambda_\alpha} [\xi_k^\alpha \xi_l^\alpha - (\xi_l^\alpha)^2]. \tag{18}$$

Averaging over all the random paths taken by the traffic packets, the steady state load at any node $j \neq l$ is $p_j^{kl} = r_j^{kl} d_j$. For the l 'th node, the load is $\mathbb{E}[X_l^{kl}] \neq p_l^{kl}$, since we defined p_l^{kl} to be zero. However, in steady state we know that the traffic flowing out of node l at any time step is unity, and this is equal to the entire load $\mathbb{E}[X_l^{kl}]$ at that time step. Therefore, in steady state, the load at the j 'th node is equal to $\Lambda_j^{kl} = d_j r_j^{kl} + \delta_{jl}$. Note that a unit of load from k to l is counted at all the nodes it passes through, as well as the source and destination nodes. Depending on how traffic is actually processed by the network, it may be appropriate to change the weightage given to the source and destination nodes.

Summing over all source destination pairs, the total steady state load at the j 'th node is

$$A_j = d_j \sum_l \sum_{k \neq l} r_j^{kl} + N - 1. \tag{19}$$

Since the first term on the right hand side of Eq. (18) is antisymmetric in k and l , only the second term contributes to $\sum_l \sum_{k \neq l} r_j^{kl}$. In the second term, we can replace the sum $\sum_{k \neq l}$ with an unrestricted sum over k , so that

$$\begin{aligned} A_j &= (N - 1) + d_j \sum_{\alpha=1}^{N-1} \frac{1}{\lambda_\alpha} \left[N \sum_l (\xi_l^\alpha)^2 - \left(\sum_l \xi_l^\alpha \right)^2 \right] \\ &= (N - 1) + Nd_j \sum_{\alpha \neq 0} \frac{1}{\lambda_\alpha}. \end{aligned} \tag{20}$$

The load A_j at any node j is linearly dependent on the degree d_j of the node. Unlike the case when traffic between any source and destination flows along the geodesic path connecting them, there is no concept of a network core. \square

Remark 2. The result $A_j - (N - 1) \propto d_j$ can be obtained directly. An outline of the proof is as follows. The traffic from node k to node l can be represented as a stream of random walkers that diffuse through the network at discrete time steps. At every time step in addition to the diffusive dynamics, a walker is introduced at node k , and all the walkers at node l are removed. Comparing with Eq. (11), the expected number of random walkers at node j at time t is equal to $p_j^{kl}(t)$. If the random walks corresponding to all source destination pairs take place simultaneously, with each walker labelled with an index corresponding to its destination, we have random walkers with N different labels moving through the network. In addition to the random walk dynamics, walkers are created and destroyed at their sources and destinations respectively. In steady state, the number of walkers created and destroyed at any time step are equal to $N - 1$ at each node, but they have different labels. If we ignore the labels on the random walkers, the creation and destruction of random walkers can be ignored. The steady state solution for $\sum_k \sum_l p_j^{kl}(t)$ is proportional to the steady state solution for a diffusion process on the graph with no sources or sinks. It is easy to verify that, in this steady state, the number of random walkers at any node is proportional to the degree of the node. Although this tells us that $[A_j - (N - 1)]/d_j$ is a constant, independent of j , it does not tell us that this constant is equal to $N \sum_{\alpha \neq 0} 1/\lambda_\alpha$.

Remark 3. If instead of using the Laplacian, L , of the graph, we had used the normalized Laplacian, \mathcal{L} , the entire proof would have proceeded as presented except that Eq. (20) would have read as follows

$$\begin{aligned} A_j &= (N - 1) + d_j \sum_{\alpha=1}^{N-1} \frac{1}{\nu_\alpha} \left[N \sum_l \left(\frac{\zeta_l^\alpha}{\sqrt{d_l}} \right)^2 - \left(\sum_l \frac{\zeta_l^\alpha}{\sqrt{d_l}} \right)^2 \right] \\ &= (N - 1) + Nd_j \sum_{\alpha \neq 0} \frac{1}{\nu_\alpha} Var\left(\frac{\zeta_l^\alpha}{\sqrt{d_l}} \right). \end{aligned} \tag{21}$$

where $0 \leq \nu_0, \dots \leq \nu_{N-1} \leq 2$ are the eigenvalues and $\{\zeta_\alpha\}$ are the corresponding orthonormal eigenvectors of \mathcal{L} and $0 \leq \lambda_0, \dots \leq \lambda_{N-1} \leq 2$ and $\{\xi_\alpha\}$ are the eigenvalues and eigenvectors of L . We note that expressions involving terms similar to the right-hand side of Eq. (21) were obtained in [16] in the context of hitting times of Markov chains, and it may be possible to obtain simpler expressions there by using the Laplacian, as we did above. Equations (20) and (21) give an interesting relationship between the spectra of the Laplacian and those of the normalized Laplacian for an arbitrary graph which we had not come across before.

Remark 4. So far we have dealt with connected undirected graphs. We point out that when the graph is directed, then assuming that steady state distribution is achieved, Remark 2 implies that the expected load $\Lambda_j = N - 1 + C\pi_j$ where C is some constant independent of the node and (π_j) is the principal eigenvector of the random walk matrix for the directed graph, which for undirected graphs is equal to (d_j) .

Remark 5. We observe that the proofs of both theorems carry through essentially unchanged if we replace the deterministic arrival of one packet at each source node for each destination node at each time step with a Poisson arrival process with a mean of one packet arrival per node per unit time for each destination node. The same is true if we replace the uniform random routing from each node to its neighbors with a more general value w_{jk}/w_j with $w_j = \sum_{l \sim j} w_{jl}$ for the probability of moving from a node j to any of its neighbors k , so long as $w_{jk} = w_{kj} \neq 0$. However, the normalized Laplacian (\mathcal{L}_{jk}) and its eigenvalues $\{\lambda_\alpha, \alpha < N\}$ in Theorem (2) are now replaced by (\mathcal{L}_{jk}^w) and its eigenvalues $\{\lambda_\alpha^w, \alpha < N\}$ where (\mathcal{L}_{jk}^w) is now the weighted normalized Laplacian [15], defined analogously as $\mathcal{L}_{jk}^w = \delta_{jk} - (1 - \delta_{jk})w_{jk}/\sqrt{w_j w_k}$ instead of $\mathcal{L}_{jk} = \delta_{jk} - (1 - \delta_{jk})/\sqrt{d_j d_k}$, see (1) in Remark 1.

2.3 Discussion

In the large- N limit, the spectral density of the Laplacian $\sum_\alpha \delta(\lambda - \lambda_\alpha)$ tends to $N\rho(\lambda)$ where $\rho(\lambda)$ is smooth. If $\rho(\lambda \rightarrow 0) = 0$, we have

$$N \sum_{\alpha \neq 0} \frac{1}{\lambda_\alpha} \rightarrow N^2 \int \frac{\rho(\lambda)}{\lambda} d\lambda \sim N^2 \tag{22}$$

for large N . The simplest example of this is when the graph Laplacian has a spectral gap in the large N limit. A more subtle case is the Erdős-Rényi model [17], where the spectral density is empirically found [18] to be close to that of an infinite regular tree whose nodes all have the same degree as the average degree of the Erdős-Rényi graph. Even though the infinite tree has a spectral gap, the corresponding Erdős-Rényi spectral density has a narrow tail extending down to $\lambda = 0$, so that there is no spectral gap [19]. However, in the next section of this paper, we find numerically that $N \sum_\alpha \lambda_\alpha^{-1} \sim N^2$ for Erdős-Rényi graphs,

presumably because the density in the tail as $\lambda \rightarrow 0$ is $\rho(\lambda \rightarrow 0) = 0$. The same result is also shown numerically for scale-free graphs.

If $\rho(\lambda \rightarrow 0)$ is not zero, $N \sum_{\alpha \neq 0} 1/\lambda_\alpha$ diverges faster than $\sim N^2$ for large N . If $\rho(\lambda \rightarrow 0)$ is finite, the spectral gap for large but finite N is proportional to $1/N$. Then $N^2 \int \rho(\lambda)/\lambda d\lambda$ diverges as $-N^2 \ln \lambda_{min} \sim N^2 \ln N$. (This is the case for the square lattice and a finite regular tree.) For hyperbolic grids $\mathbb{H}_{p,q}$ (where p and q are positive integers satisfying $(p - 2)(q - 2) > 4$), which are infinite regular planar graphs with constant degree q and p -sided polygons as faces, we show numerically in the next section of this paper that $N \sum \lambda_\alpha^{-1} \sim N^2 \ln N$.

The maximum congestion in the network is, up to an additive constant, equal to the product of $N \sum 1/\lambda_\alpha$ and d_{max} . The large- N dependence of the latter depends on the degree distribution, e.g growing as $\sim \ln N$ for Erdős-Rényi graphs and as a power of N for scale-free networks.

The average time τ that a packet spends in the network is obtained from the equation $N(N - 1)\tau = \sum_j A_j$, from which

$$\tau = \frac{\sum_j d_j}{N - 1} \sum_{\alpha=1}^{N-1} \frac{1}{\lambda_\alpha} + 1 \rightarrow \bar{d} \sum_{\alpha=1}^{N-1} \frac{1}{\lambda_\alpha} \tag{23}$$

in the large N limit, where \bar{d} is the average degree of nodes in the graph. If $\sum \lambda_\alpha^{-1} \sim N$, the average sojourn time in the graph is $O(N)$. To express this in terms of the diameter of the graph instead of the number of nodes, we have to know how the diameter grows as N is increased; for small world graphs, τ grows exponentially as the diameter of the graph is increased. Exponential growth implies that a shortest-path walk starting at a site k and aimed at site l can reach destination l exponentially faster on average than the random walk.

3 Numerical Results

In this section, we present numerical results for a few prototypical graph models: the Erdős-Rényi random graphs in various regimes, the Barabási-Albert model of preferential attachment [20, 21], and hyperbolic grids.

Because of its zero eigenvalue, the matrix L is not invertible. We define the matrix $M = L + P$, where $P_{ij} = 1/N$. Then P is a projection operator: $P \sum_\alpha c_\alpha \xi^\alpha = c_0 \xi^0$. Therefore

$$M \sum_\alpha c_\alpha \xi^\alpha = \sum_\alpha (\lambda_\alpha + \delta_{\alpha 0}) c_\alpha \xi^\alpha. \tag{24}$$

Therefore M is an invertible matrix, with $\text{Tr}[M^{-1}] = \sum_\alpha (\lambda_\alpha + \delta_{\alpha 0})^{-1}$, which is equal to $1 + \sum_{\alpha \neq 0} \lambda_\alpha^{-1}$. We have to numerically evaluate $\text{Tr}[M^{-1}] - 1$.

Figure 1 shows the results for $N \sum \lambda_\alpha^{-1}$ for the Erdős-Rényi model as N is increased. Two cases are considered: when the average nodal degree d_a is 2 and 4. Since $d_a > 1$, there is a giant component in each graph, containing an N -independent fraction of the nodes in the large- N limit. All the other nodes

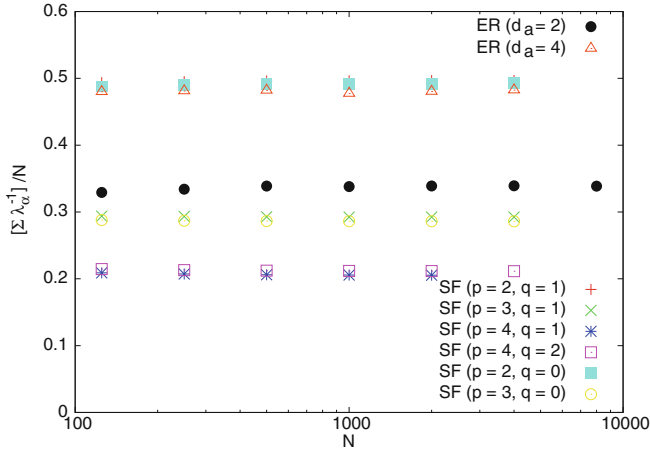


Fig. 1. Plot of $[\sum_{\alpha \neq 0} \lambda_\alpha^{-1}]/N$ versus N for the Erdős-Rényi model with average nodal degree of 2 and 4. (For the first of these, the vertical axis is scaled by a factor of 0.25 to fit in the figure.) Also shown are the results for scale free networks, where each node is born with p edges that link it to preexisting nodes, and the probability of linking to a preexisting node is proportional to its degree with an offset of q ; the results for various values of (p, q) are shown. The curves are flat for all the cases, demonstrating that $N \sum \lambda_\alpha^{-1} \sim N^2$.

are in components whose size does not diverge as N is increased. Since we are considering graphs with a single component in this paper, only the giant component of each graph is retained. This means that the actual number of nodes in the graph is a d_a -dependent fraction of the N shown in Fig. 1, but this does not affect the functional form of large- N behavior. Each point shown in the figure comes from averaging over eighty random graphs. We see that $N \sum \lambda_\alpha^{-1} \sim N^2$.

Figure 1 also shows results for scale free networks. Following the extension of Ref. [21] of the original model of Ref. [20], nodes enter the network one by one, with each node born with p edges that link it to pre-existing nodes; the probability of linking to any preexisting node is proportional to $d - q$ if its degree is d , where q is a parameter of the model. The figure shows results for $(p, q) = (2, 0), (3, 0), (2, 1), (3, 1), (4, 1)$ and $(4, 2)$. As with the Erdős-Rényi graphs, each point in the figure comes from averaging over eighty random graphs. Once again, $N \sum \lambda_\alpha^{-1} \sim N^2$.

The first panel of Fig. 2 shows the results for $N \sum \lambda_\alpha^{-1}$ for the hyperbolic grid $\mathbb{H}_{3,7}$. The data clearly show that $N \sum \lambda_\alpha^{-1} \sim N^2 \ln N$.

In the Erdős-Rényi model, if $d_a = c \ln N$ instead of being independent of N , there is a phase transition in the behavior of the model when c is increased to 1: the fraction of the nodes in the giant component approaches 1. The behavior of graphs constructed using this model is very different in this regime. The second panel of Fig. 2 shows the results for $N \sum \lambda_\alpha^{-1}$ when $d_a = \ln N$. We see that $N \sum \lambda_\alpha^{-1}$ grows *slower* than N^2 for large N . Although the data are

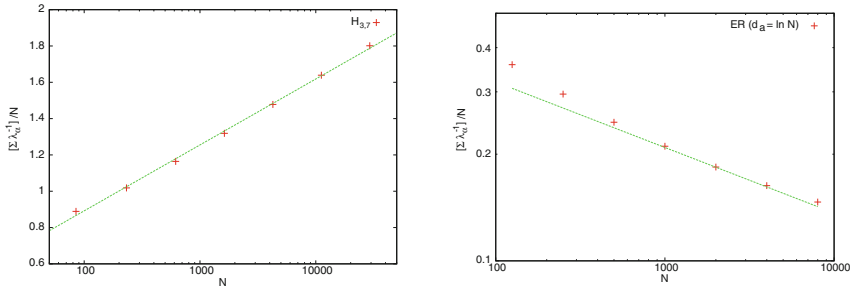


Fig. 2. Plot of $[\sum_{\alpha \neq 0} \lambda_{\alpha}^{-1}] / N$ versus N for a) the hyperbolic grid, with seven triangles meeting at every node. All the nodes that are less than some distance r from a center node are included; N increases with r . With the x -axis on a logscale, the straight line fit demonstrates that $N \sum \lambda_{\alpha}^{-1} \sim N^2 \ln N$. b) the Erdős-Rényi model with the average degree of the nodes equal to $\ln N$. The straight line shown corresponds to $0.75N^{-0.185}$.

not conclusive, they suggest a $\sim N^{2-\alpha}$ form. As with the other random graph models, each point in the figure is obtained by averaging over eighty random graphs.

As mentioned earlier in this paper, the maximum load for all the nodes in a graph consists of—apart from an additive term—the product of $N \sum_{\alpha} \lambda_{\alpha}^{-1}$ and the highest nodal degree in the graph. For scale free graphs, if the probability of a node having a degree d scales as $p(d) \sim d^{-\gamma}$ for large d , the highest nodal degree in a graph with N nodes scales as $N^{1/(\gamma-1)}$ for large N .

4 Conclusions

We showed for the uniform multicommodity flow problem on an arbitrary connected graph under random routing, the mean load (or congestion) at each node of the graph exists, is unique and derived an explicit expression for it in terms of the spectrum of the graph Laplacian. Using this explicit expression, we obtained analytical estimates for the mean load for hypercubic lattices and regular trees in the large-size regime using their known spectral densities and computed numerically the mean load for the Erdős-Rényi random graphs, the scale-free Barabási-Albert preferential attachment graphs and hyperbolic grids.

Acknowledgements. This work of Onuttom Narayan and Iraj Saniee was supported by grants FA9550-11-1-0278 and 60NANB10D128 from AFOSR and NIST.

References

1. Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. *Canadian J. Math.* **8**, 399–404 (1956)
2. Elias, P., Feinstein, A., Shannon, C.: A note on the maximum flow through a network. *IEEE Trans. Inf. Theory* **2**, 117–119 (1956)

3. Hu, T.C.: Multicommodity network flows. *Op. Res.* **11**, 344–360 (1963)
4. Lomonosov, M.V.: Combinatorial approaches to multiflow problems. *Discrete Appl. Math* **11**, 1–94 (1985)
5. Shahrokhi, F., Matula, D.W.: The maximum concurrent flow problem. *J. ACM* **37**, 318–334 (1990)
6. Papernov, B.A.: Feasibility of multicommodity flows (in Russian). In: Friedman, A. (ed.) *Studies in Discrete Optimization*, pp. 230–261. Idzat. “Nauka”, Moscow (1976)
7. Okamura, H., Seymour, P.D.: Multicommodity flows in planar graphs. *J. Combinatorial Theory Series B* **31**, 75–81 (1981)
8. Minou, M.: Network synthesis and optimum network design problems: models, solution methods and applications. *Networks* **19**, 313–360 (1989)
9. Magnanti, T.L., Wong, R.T.: Network design and transportation planning: models and algorithms. *Transp. Sci.* **18**, 1–55 (1984)
10. Applegate, D., Cohen, E.: Making intra-domain routing robust to changing and uncertain traffic demands: understanding fundamental tradeoffs. In: *ACM Sigcomm 2003*, pp. 313–324 (2003)
11. Narayan, O., Saniee, I.: Large-scale curvature of networks. *Phys. Rev. E* **84**, 066108 (2011)
12. Jonckheere, E.A., Lou, M., Bonahon, F., Baryshnikov, Y.: Euclidean versus hyperbolic congestion in idealized versus experimental networks. *Internet Math.* **7**, 1–27 (2011)
13. Newman, M.E.J.: *Networks: An Introduction*. Oxford University Press, Oxford (2010)
14. Newman, M.E.J.: A measure of betweenness centrality based on random walks. *Soc. Netw.* **27**, 39–54 (2005)
15. Chung, F.R.K.: *Spectral Graph Theory*. American Mathematical Society, Providence (1997)
16. Lovasz, L.: Random walks on graphs: a survey. In: Miklós, D., Sós, V.T., Szönyi, T. (eds.) *Combinatorics—Paul Erdős is Eighty 2*, pp. 1–46. Janos Bolyai Mathematical Society, Keszthely (1993)
17. Erdős, P., Rényi, A.: On random graphs. *Publicationes Mathematicae* **6**, 290–297 (1959)
18. Narayan, O., Saniee, I., Tucci, G.H.: Lack of hyperbolicity in asymptotic Erdős-Rényi sparse random graphs. *Internet Math.* **11**, 277–288 (2015)
19. Samukhin, A.N., Dorogovtsev, S.N., Mendes, J.F.F.: Laplacian spectra of complex networks and random walks on them: are scale-free architectures really important? *Phys. Rev. E* **77**, 036115 (2008)
20. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* **286**, 509–512 (1999)
21. Dorogovtsev, S.N., Mendes, J.F.F., Samukhin, A.N.: Structure of growing networks: exact solution of the Barabasi-Albert model. *Phys. Rev. Lett.* **85**, 4633–4636 (2000)



Strongly Connected Components in Stream Graphs: Computation and Experimentations

Léo Rannou^{1,2(✉)}, Clémence Magnien¹, and Matthieu Latapy¹

¹ Sorbonne Université, CNRS, LIP6, 75005 Paris, France
leo.rannou@lip6.fr

² Thales SIX, Theresis, 1 av. Augustin Fresnel, 91120 Palaiseau, France

Abstract. Stream graphs model highly dynamic networks in which nodes and/or links arrive and/or leave over time. Strongly connected components in stream graphs were defined recently, but no algorithm was provided to compute them. We present here several solutions with polynomial time and space complexities, each with its own strengths and weaknesses. We provide an implementation and experimentally compare the algorithms in a wide variety of practical cases. In addition, we propose an approximation scheme that significantly reduces computation costs, and gives even more insight on the dataset.

Keywords: Stream graphs · Link streams · Temporal graphs · Temporal networks · Dynamic graphs · Connected components · Algorithms

Connected components are among the most important concepts of graph theory. They were recently generalized to stream graphs [18], a formal object that captures the dynamics of nodes and links over time. Unlike other generalizations available in the literature, these generalized connected components *partition* the set of temporal nodes. This means that each node at each time instant is in one and only one connected component. This makes these generalized connected components particularly appealing to capture important features of objects modeled by stream graphs. However, computation of connected components in stream graphs has not been explored yet. Therefore, up to this date, they remain a formal object with no practical use. In addition, the algorithmic complexity of the problem is unknown, as well as the insight they may shed on real-world stream graphs of interest.

After introducing key notations and definitions (Sect. 1), we present two algorithms for strongly connected components, together with their complexity (Sect. 2). We then apply these algorithms to several large-scale real-world datasets and demonstrate their ability to describe such datasets (Sect. 3). We also show that their performances may be improved greatly at the cost of reasonable approximations.

1 The Stream Graph Framework

Given any two sets A and B , we denote by $A \otimes B$ the set of pairs ab such that $a \in A, b \in B$ and $a \neq b$. Couples are ordered, while pairs are unordered: $(a, b) \neq (b, a)$ while $ab = ba$.

A **stream graph** $S = (T, V, W, E)$ is defined [18] by a finite set of nodes V , a time interval $T \subseteq \mathbb{R}$, a set of temporal nodes $W \subseteq T \times V$, and a set of links $E \subseteq T \times V \otimes V$ such that $(t, uv) \in E$ implies $(t, u) \in W$ and $(t, v) \in W$.

For any u and v in $V, T_u = \{t, (t, u) \in W\}$ denotes the set of time instants at which u is present, and $T_{uv} = \{t, (t, uv) \in E\}$ the set of time instants at which u and v are linked together. We assume that both T_u and T_{uv} are unions of a finite number of disjoint closed intervals (possibly singletons) of T .

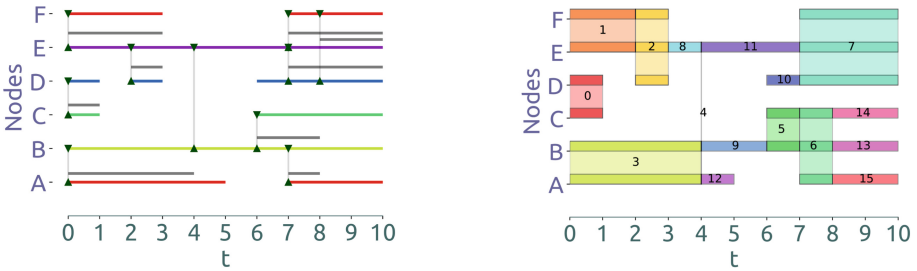


Fig. 1. (Left) An example of stream graph. We display time $T = [0, 10]$ on the horizontal axis and nodes $V = \{A, B, C, D, E, F\}$ on the vertical one. We represent each node segment by a colored horizontal segment, with one color per node; and each link segment in grey by a vertical line between the two involved nodes at the link segment starting time, and an horizontal line from this time to its ending time. (Right) The 16 strongly connected components of the stream graph.

We call *node segment* a couple $([b, e], u)$ such that $[b, e]$ is a segment that is not included in any other segment of T_u , and we denote by \overline{W} the set of all node segments in W . We say that b is an *arrival* of u , and e a *departure*. We denote by $N = |\overline{W}|$ the number of node segments in the stream. Likewise, we call *link segment* a couple $([b, e], uv)$ such that $[b, e]$ is not included in any other segment of T_{uv} , and by \overline{E} the set of all link segments in E . We say that b is an *arrival* of uv , and e a *departure*. We denote by $M = |\overline{E}|$ the number of link segments in the stream. We call all time instants that correspond to a node or link arrival or departure an *event time*. There are at most $2 \cdot N + 2 \cdot M$ event times. Notice that the intervals considered above may be singletons. Then, $b = e$ and $[b, e] = \{b\} = \{e\}$. See Fig. 1 for an illustration.

The induced graph $G(S) = (V(S), E(S))$ is defined by $V(S) = \{v, T_v \neq \emptyset\}$ and $E(S) = \{uv, \exists t, (t, uv) \in E\}$. We denote by $n = |V(S)|$ and $m = |E(S)|$ its number of nodes and links, respectively. We denote by $G_t = (V_t, E_t)$ the graph such that $V_t = \{v, (t, v) \in W\}$ and $E_t = \{uv, T_{uv} \neq \emptyset\}$. We denote by G_t^- the

graph that corresponds to the nodes and links present between the event time just before t and t : $G_t^- = (V_t^-, E_t^-)$ where $V_t^- = \{v, \exists t' \neq t, [t', t] \subseteq T_v\}$ and $E_t^- = \{uv, \exists t' \neq t, [t', t] \subseteq T_{uv}\}$.

We consider in input a time-ordered sequence of node or link arrivals or departures. We maintain the set of present nodes and links at the current time instant t , *i.e.* the graph G_t , and we store their latest arrival time seen so far. This has a $\Theta(N + M)$ time and $\Theta(n + m)$ space cost for the whole processing of input data. Therefore, these worst-case complexities are lower bounds for our algorithms.

2 Strongly Connected Components

A **strongly connected component** of $S = (T, V, W, E)$ is a maximal subset $I \times X$ of W such that I is an interval of T and X is a connected component of G_t for all t in I . It is denoted by (I, X) . The set of all strongly connected components of S is a partition of W [18]. See Fig. 1 for an illustration.

Notice that some component time intervals are closed, some are open and some are a combination of the two. For instance, $([0, 1], \{C, D\})$ is a closed component, $(]4, 6[, \{B\})$ is an open one, $([6, 7[, \{B, C\})$ is a left-closed and right-open one, and $([4, 4], \{A, B, C\})$ is a closed and instantaneous component. Since the time intervals of components may be open or closed, we introduce the notation $\langle b, e \rangle$ to indicate an interval that can be either open or closed on its extremities. This interval contains $]b, e[$ and may or may not contain b and/or e . We will also use mixed notation: $\langle b, e \rangle$ for instance designates an interval that may or may not contain b , but does contain e .

The number of strongly connected components is in $\Theta(N + M)$, because there can be one component per node segment, and each link segment may induce up to four components. Indeed, each beginning of a link segment may correspond to the beginning of two components: one instantaneous at the link segment beginning and one that starts just after; and each link segment ending may correspond to the beginning of two connected components if the corresponding component becomes disconnected. Explicitly writing a component to the output is done in linear time with respect to its number of nodes, in $\Omega(N + n \cdot M)$.

2.1 Direct Approach

One may compute strongly connected components directly from their definition, by processing event times in increasing order and by maintaining the set of strongly connected components that begin before or at current event time, and end after it. We represent each such component as a couple $(\langle b, C \rangle)$, meaning that it starts at b (included or not) and involves nodes in C .

More precisely, we start with a set \mathcal{C} containing $([\alpha, C)$ for each connected component C of the graph G_α at the first event time α . Then, for each event time $t > \alpha$ in increasing order we consider the connected components of G_t^- . For each such component C , if there is no component $(\langle b, X \rangle)$ with $X = C$ in

\mathcal{C} then we add $(\lceil t', C)$ to \mathcal{C} , where t' is the event time preceding t . For each element $(\langle b, X)$ of \mathcal{C} , if X is not a connected component of G_t^- , then we remove it from \mathcal{C} and we output $(\langle b, t', X)$. We then turn to the connected components of G_t : for each such component C , if there is no component $(\langle b, X)$ with $X = C$ in \mathcal{C} then we add $(\lceil t, C)$ to \mathcal{C} ; and for each element $(\langle b, X)$ of \mathcal{C} , if X is not a connected component of G_t , then we remove it from \mathcal{C} and we output $(\langle b, t, X)$. Finally, when the last event time $t = \omega$ is reached, we output $(\langle b, \omega, X)$ for each element $(\langle b, X)$ of \mathcal{C} .

Clearly, this algorithm outputs all strongly connected components of the considered stream graph. Computing the connected components of each graph is in $O(n + m)$ time and space. The considered set families (the graph connected components, as well as the elements of \mathcal{C}) form partitions of V . Therefore, their storage and all set comparisons processed for each event time have a cost in $O(n)$ time and space. There are $O(M + N)$ event times, therefore, the time complexity of this method is $O((N + M) \cdot (n + m))$, and it needs $O(n + m)$ space.

Without changing its time complexity, this algorithm may be improved by ignoring event times t such that all events occurring at t are link arrivals between nodes already in the same connected component. However, one still has to compute graph connected components at each event time with link departures. Therefore, this improvement is mostly appealing if many link departures occur at the same event times.

More generally, the approach above is efficient only if many events (node and/or links arrivals and/or departures) occur at each event time. Then, many connected components may change at each event time, and computing them from scratch makes sense. Instead, if only few events occur at most event times, managing each event itself and updating current connected components accordingly is appealing.

This leads to the following algorithm, which starts with an empty set \mathcal{C} , considers each event time t in increasing order, and performs the following operations.

1. For each node segment $([b, e], u)$ such that $b = t$ (node arrival), add $(\lceil b, \{u\})$ to \mathcal{C} .
2. For each link segment $([b, e], uv)$ such that $b = t$ (link arrival), let $C_u = (\langle b_u, X_u)$ and $C_v = (\langle b_v, X_v)$ be the elements of \mathcal{C} such that $u \in X_u$ and $v \in X_v$; if $C_u \neq C_v$ then replace C_u and C_v by $(\lceil t, X_u \cup X_v)$ in \mathcal{C} . Then: if $\langle b_u \neq \lceil t$ then output $(\langle b_u, t, X_u)$; if $\langle b_v \neq \lceil t$ then output $(\langle b_v, t, X_v)$.
3. Let $G'_t = G_t$; then for each link segment $([b, e], uv)$ such that $e = t$ (link departure), let $C_v = C_u = (\langle b_u, X_u)$ be the element of \mathcal{C} such that $u \in X_u$ and $v \in X_u$; remove the link uv from G'_t ; if there is no path between u and v in G'_t then replace C_u by $C'_u = (\lceil t, X'_u)$ and $C'_v = (\lceil t, X'_v)$ in \mathcal{C} where X'_u and X'_v are the connected components of u and v in G'_t , respectively; if $\langle b_u \neq \lceil t$ then output $(\langle b_u, t, X_u)$.
4. For each node segment $([b, e], u)$ such that $e = t$ (node departure), let $C_u = (\langle b_u, X_u)$ be the element of \mathcal{C} such that $u \in X_u$; remove C_u from \mathcal{C} ; if $\langle b_u \neq \lceil t$ then output $(\langle b_u, t, \{u\})$.

We call this algorithm *SCC Direct*. It clearly outputs the strongly connected components of the considered stream, like the previous algorithm. It performs $2(M + N)$ of the steps above, corresponding to N node arrivals and departures and M link arrivals and departures. One easily deals with node arrivals and departures in constant time. If a link arrival induces a merge between two components, computing their union is in $O(n)$, as is outputting both components if needed. Thus the complexity for link arrival steps is in $O(M \cdot n)$. Each link departure calls for a computation of the connected components of a graph, and writing a component to the output is in $O(n)$. Thus the complexity for link departure steps is in $O(M \cdot (m + n))$. We obtain a total time complexity in $O(M \cdot (m + n) + N)$. The space complexity is still in $O(n + m)$ as above.

2.2 Fully Dynamic Approach

The SCC Direct algorithm presented above is strongly related to one of the most classical algorithmic problems in dynamic graph theory, called fully dynamic connectivity [2, 9, 10, 13–15, 26], which aims at maintaining the connected components of an evolving graph. Considering a sequence of link additions and removals, dynamic connectivity algorithms maintain a data structure able to tell if two nodes are in the same connected components (*query* operation) and to merge or split connected components upon link addition or removal (*update* operation).

This data structure and the corresponding operations can be used in the above algorithm: we can use the data structure to store \mathcal{C} , the set of current connected components (we also need to store the beginning time of each component, which has negligible cost). Then, at each link arrival or departure, we can use the query operation to test whether the two nodes are in the same component or not, and the update operation to add or remove the current link to the data structure, while keeping an up-to-date set of connected components. When we observe a node appearance it is necessarily isolated, so we have to add the current time to its component. All the other steps (mainly, writing the output) are unchanged. We call this algorithm *SCC FD*.

Several methods efficiently solve the dynamic connectivity problem, the key challenge being to know if updates and queries may be performed in $O(\log(n))$ time, where n is the number of nodes in the graph. Current exact solutions perform updates in $O\left(\sqrt{\frac{n \cdot (\log \log(n))^2}{\log(n)}}\right)$ worst time [15], or in $\frac{\log^2(n)}{\log \log(n)}$ amortized worst time [26]. Probabilistic (exact or approximate) methods perform even better, but they remain above the $O(\log(n))$ time cost [9, 10, 14].

It is well acknowledged that these algorithmic time and space complexities hide big constants, and that the underlying algorithms and data structures are very intricate. As a consequence, implementing these algorithms is an important challenge in itself [2, 13], and the results above should be considered as theoretical bounds. In practice, the implemented algorithms typically have $O(\log(n)^3)$ amortized time and linear space complexities, still with large constants [2, 13].

In SCC FD, we perform $O(M)$ updates and queries, which leads to a $O(M \cdot \text{polylog}(n))$ overall time cost for these operations, with any of the polylog dynamic connectivity algorithms cited above. This cost is dominated by the cost of outputting the results, which is in $O(M \cdot n)$. An additional N factor is needed to deal with node arrivals and departures. Hence, we obtain a total time in $O(M \cdot n + N)$. The space cost of dynamic connectivity methods is in $O(m + n \cdot \log n)$, and we do not store significantly more information.

This algorithm is particularly appealing if large connected components are quite stable, *i.e.* if most largest strongly connected components in the stream have a long duration. Indeed, in this case, fully dynamic update operations are much faster than updates used in SCC Direct, and the output is much smaller than the maximum $\Omega(N + M \cdot n)$ bound. The cost of SCC FD is then dominated by fully dynamic operations, and its time complexity is reduced to $O(M \cdot \text{polylog}(n))$.

3 Experiments and Applications

In this section, we conduct thorough experiments with several real-world datasets and our different algorithms. SCC Direct was significantly faster, and only SCC Direct was able to perform the computation in central memory of large-scale datasets (several dozens of millions of link segments). We publicly provide Python 3 implementations of our algorithms in the Straph library [23].

3.1 Datasets

First notice that most available datasets record instantaneous interactions only, either because of periodic measurements, or because only one timestamp is available. In such situations, one resorts to δ -analysis [18]: one considers that each interaction lasts for a given duration δ . This transforms a dataset into a stream graph $S = (T, V, W, E)$ in which all link segments last for at least δ , and all links in D separated by a delay lower than δ lead to a unique link segment. Nodes are considered as present only when they have at least one link.

In order to explore the performances of our algorithms in a wide variety of situations, we considered 14 publicly available datasets that we shortly present below. Their key stream graph properties are given in Table 1, together with the value of δ we used. It either corresponds to a natural value underlying the dataset or is determined by the original timestamp precision.

UC Message (UC) [17] is a capture of messages between University of California students in an online community. **High School 2012 (HS 2012)** [6] is a sensor recording of contacts between students of 5 classes during 7 days in a high school in Marseille, France in 2012. **Digg** [17] is a set of links representing replies of Digg website users to others. **Infectious** [12] is a recording of face-to-face contacts between visitors of an exhibition in 2009, Dublin. **Twitter Higs (Twitter)** [4, 19] is a recording of all kinds of twitter activity for one week around the discovery of the Higgs boson in 2012. **Linux Kernel mailing list (Linux)** [17]

Table 1. Key features of the real-world stream graphs we consider, ordered with respect to their number M of link segments (K indicates thousands, M millions).

	δ	n	m	$ T $	N	M
UC	1 h	2K	14K	189d	43K	34K
HS 2012	60 s	327	6K	4d	48K	46K
Digg	1 h	30K	85K	14.5d	110K	86K
Infectious	60 s	11K	45K	80d	85K	133K
Twitter	600 s	304K	452K	7d	543K	488K
Linux	10 h	27K	160K	8y	450K	544K
Facebook	10 h	46K	183K	4.3y	957K	588K
Epinions	10 h	132K	711K	2.6y	404K	743 K
Amazon	1 h	2.1M	5.7M	9.5y	9.9M	5.8M
Youtube	24 h	3.2M	9.4M	226d	6.7M	9.4M
Movielens	1 h	70K	10M	14y	8.5M	10M
Wiki	1 h	2.9M	8.1M	14.3y	18.3M	14.5M
Mawilab	2 s	940K	9.1M	902 s	17M	18.8M
Stackoverflow	10 h	2.6M	28.2M	7.6y	30M	33.5 M

represents the email replies between users on this mailing-list. **Facebook wall posts (Facebook)** [25] represents messages exchanged between Facebook users, through their walls. **Epinions** [17] is a set of timestamped trust and distrust link creations on Epinions, an online product rating site. **Amazon** [17] contains product ratings on Amazon. **Youtube** [20] is a social network of YouTube users and their friendship connections. **Movielens** [8] contains movie ratings by users of the Movielens site. **Wiki Talk En (Wiki)** [17] is a recording of discussions between contributors to the English Wikipedia. **Mawilab 2020-03-09 (Mawilab)** [5] is a 15 min capture of network traffic on a backbone trans-pacific router in Japan on March 3, 2020. Each link represents a packet exchanged between two internet addresses. **Stackoverflow** [19,22] is a recording of interactions on the stack overflow web site.

3.2 Algorithm Performances

Figure 2 presents the time cost for each dataset, and show a strong relation between the number of link segments, the number of connected components, and computation time. Notice however that *Wiki* and *Mawilab* have similar numbers of link and node segments but *SCC Direct* is several order of magnitude faster on *Wiki*. This difference comes from their quite different structure regarding connected components: *Mawilab* has more than 21M SCC involving at least 30K nodes, whereas *Wiki* has only 2K such SCC. As explained in Sect. 2, the computational cost of *SCC Direct* mainly depends on the number of nodes in each SCC, which is observed in this experiment.

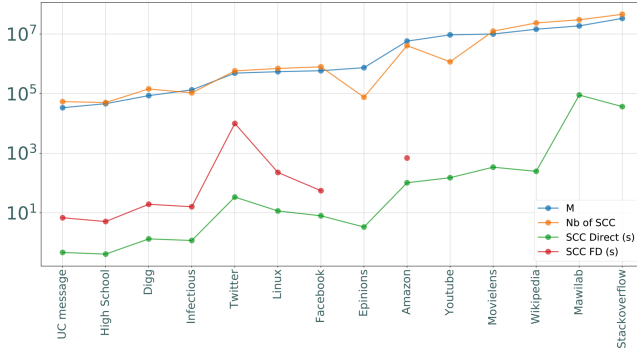


Fig. 2. Time cost of *SCC Direct* and *SCC FD* in seconds, along with the number M of link segments and the number of strongly connected components, for each considered stream (horizontal axis, ordered with respect to M).

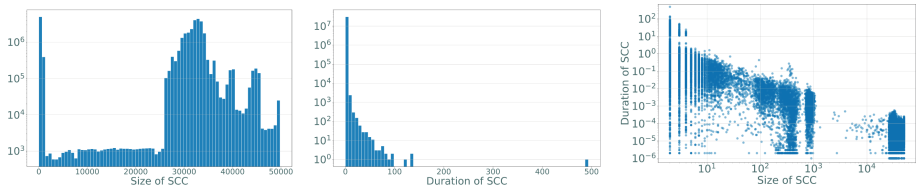


Fig. 3. Distribution of the size (left) and duration (middle) of SCC in *Mawilab* dataset. Duration of each SCC as a function of its size, in log-log scales (right).

3.3 Connectedness Analysis of IP Traffic

We take the *MawiLab* IP traffic capture as a typical instance of large real-world datasets modeled by a stream graph, and we use it to illustrate the relevance of connected component analysis. The stream has 30,062,184 such components, with no giant one. Given $C = ((b, e), X)$ we call the number of involved nodes $|X|$ its size, and the length of its time interval $e - b$ its duration.

In Fig. 3, we display the strongly connected component size and duration distributions as well as the duration of components as a function of their size. Clearly, component size and duration are not linearly dependant. **No component significantly stands out of the crowd: there is no component with both a long duration and a large size.** Instead, large strongly connected components have a very short duration, and, conversely, long components have a small size. For instance, all components involving at least $2K$ nodes have a duration lower than $1e-3$ s, and all components that last for more than two seconds involve less than ten nodes. The largest component (in terms of number of nodes) involves 49,791 nodes (only 5.3% of the whole), and lasts for $8.2e-5$ s (only $9.1e-6\%$ of the whole).

More generally, these plots show that there are many strongly connected components with very short duration: 90% last less than 0.14 s. These compo-

nents are due to the *frontier effect*, that we define as follows. Consider a set X of nodes, and assume that link segments that start close to a given time b and end close to a given time e connect them. However, they all start at different times and end at different times. This leads to a connected component $([b', e'], X)$, with b' close to b and e' close to e , but also to many short strongly connected components that both start and end close to b , or close to e . These components make little sense, if any, but they account for a huge fraction of all strongly connected components, and so they have a crucial impact on computation time as explained in the previous section. We show below how to get rid of them while keeping crucial information.

3.4 Approximate Strongly Connected Components

The fact that link segments start and end at slightly different times induces many strongly connected components of very low duration, that have little interest. We, therefore, propose to consider the following approximation of the stream graph $S = (T, V, W, E)$. Given an approximation parameter $\Delta < \delta$ and any time t in T , we define $\lfloor t \rfloor_\Delta$ as $\Delta \cdot \lfloor \frac{t}{\Delta} \rfloor$ and $\lceil t \rceil_\Delta$ as $\Delta \cdot \lceil \frac{t}{\Delta} \rceil$. We then define $S_\Delta = (T, V, W_\Delta, E_\Delta)$ where $W_\Delta = \cup_{([b,e],v) \in \overline{W}} [\lfloor b \rfloor_\Delta, \lfloor e \rfloor_\Delta] \times \{v\}$ and $E_\Delta = \cup_{([b,e],uv) \in \overline{E}} [\lfloor b \rfloor_\Delta, \lfloor e \rfloor_\Delta] \times \{uv\}$. In other words, we replace each node segment $([b, e], v)$ by a shorter node segment that starts at the first time after b and ends at the last time before e which are multiple of Δ . We proceed similarly with link segments.

First notice that S_Δ is an approximation of S , in the sense that S_Δ may be computed from S , but not the converse. In addition, each node or link segment in S lasts at least δ , and since Δ is lower than δ , no node or link segment disappears when S is transformed into S_Δ ; only their starting and ending times change. In addition, S_Δ is included in S : $W_\Delta \subseteq W$ and $E_\Delta \subseteq E$. This has an important consequence: all paths in S_Δ are also paths in S , and so the approximation does not create any new reachability relation. It, therefore, preserves key information contained in the original stream.

Let us first observe the effect of the approximation on strongly connected components in Fig. 4. The number of components rapidly drops from its initial value of 30 millions (for $\Delta = 0$, *i.e.* no approximation) to less than 6 millions for $\Delta = \delta/10^3 = 0.002$. Its decrease is much slower when Δ grows further, which indicates that the stream does not anymore contain an important number of irrelevant components due to the *frontier effect*. As expected, this has a strong impact on computation time, which we also display; it also very rapidly drops, from more than one day to less than one hour, making computations on such large-scale datasets much quicker.

Figure 5 presents the effect of Δ on size, duration and span distributions of strongly connected components. For $\Delta = \delta/10^4$, we notice that while the number of components has decreased by half only fifty percent of them involve more than 30K nodes. Furthermore, as Δ increases, the number of components tends to be stable (Fig. 4) but the number of components involving more than 30K nodes

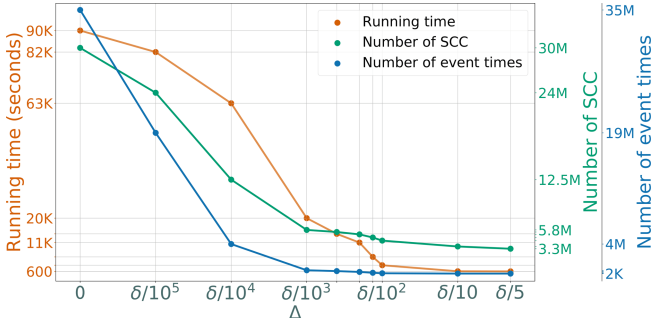


Fig. 4. Running time of *SCC Direct*, number of SCC and number of event times in MawiLab, as a function of Δ (here, $\delta = 2s$).

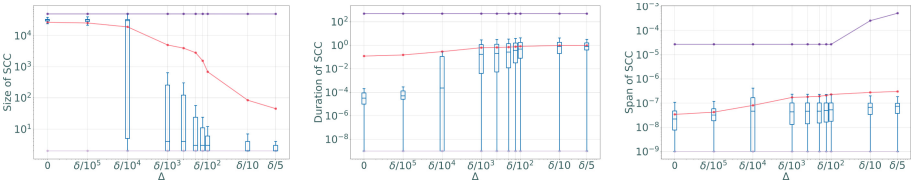


Fig. 5. Box plots representing the distribution of the size (left), duration (middle) and span (right) of strongly connected components in *Mawilab*, for various values of Δ (here, $\delta = 2s$). We indicate the mean, minimal, and maximal values with dots connected by horizontal lines, as well as the median and percentiles with vertical boxes.

continues to drop. This explains the differences observed in the execution time of *SCC Direct* (Fig. 4) and confirms that the approximation eliminates most very short connected components, but not all: the ones which are not due to the frontier effect are preserved, another wanted feature.

3.5 Application to Latency Approximation

Although the approximation above has a strong impact on the number of strongly connected components, it preserves key information of the stream. We illustrate this by considering one of the most widely studied features of these objects: the latency between nodes [16, 27]. Given two nodes u and v in a stream graph $S = (T, V, W, E)$, the latency from u to v , denoted by $\ell(u, v)$, is the minimal time needed to reach v from u by following links of S in a time-respecting manner, and taking into account node dynamics, see [18] for details.

Notice that latencies in S_Δ are necessarily larger than or equal to latencies in S , since paths in S_Δ are also paths in S . Therefore, latencies in S_Δ are upper bounds of latencies in S , and we show below that they are actually quite close.

Figure 6 displays the average difference between latencies in S and S_Δ as a function of Δ for the Mawi dataset: $\frac{\sum_{u,v \in V, u \neq v} \ell_\Delta(u,v) - \ell(u,v)}{n \cdot (n-1)}$. It also dis-

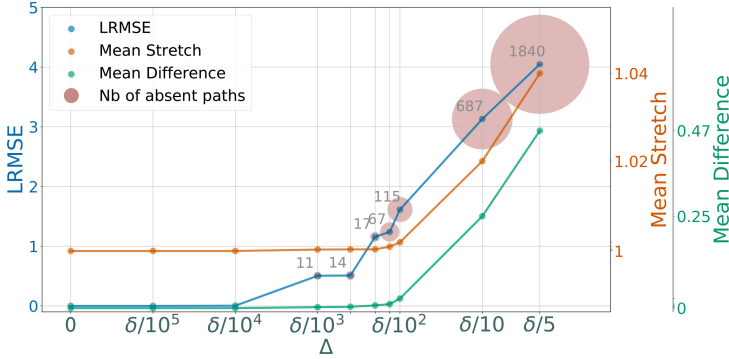


Fig. 6. Evolution of the LRMSE, the average difference between latencies and the average latency stretch with respect to Δ in *Mawilab*. We indicate the number of missing paths and represent it as a disk of area proportional to this number.

plays the average latency stretch $\frac{\sum_{u,v \in V, u \neq v} (\ell_\Delta(u,v)+1)/(\ell(u,v)+1)}{n \cdot (n-1)}$ and the latency root mean square error: $LRMSE(S, S_\Delta) = \sqrt{\frac{\sum_{u,v \in V, u \neq v} (\ell(u,v) - \ell_\Delta(u,v))^2}{n(n-1)}}$. The figure also indicates the number of node pairs that were reachable in S but became unreachable in this approximation. It appears that latencies are not significantly impacted by approximation, thus confirming that S_Δ , despite its reduced number of strongly connected components, captures key information available in S . More precisely, only 11 temporal paths disappear for $\Delta = \delta/10^3$ and 115 disappear for $\Delta = \delta/10^2$, among a total number of 2,888,917. The over-estimate of latencies is very small, with a *LRMSE* of 0.51 and 1.61, respectively. This has important consequences. For instance, one may compute latencies in S_Δ from its strongly connected components, which are much easier to compute and store than the ones of S , and obtain this way fast and accurate upper bounds (or approximations) of latencies in S , like we did here for the *Mawilab* dataset.

4 Related Work

We focus here on connected components defined in [18], but other notions of connected components in dynamic graphs have been proposed. Several rely on the notion of reachability, which, in most cases, induces components that may overlap and are NP-hard to enumerate, see for instance [3, 7, 11, 21]. This makes them quite different from the connected components considered here.

Akradi and Spirakis [1] study and propose an algorithm for testing whether a given dynamic graph is connected at all times during a given time interval. If it is not connected, their algorithm looks for large connected components that exist for a long duration. Vernet *et al.* [24] propose an algorithm for computing all sets of nodes that remain connected for a given duration, and that are not dominated by other such sets. Unlike our work, these papers do not partition the set of temporal nodes.

Finally, observing the size of largest components is classical, and Nicosia *et al.* [21] study them in time-varying graphs, with a connectivity based on reachability through temporal paths. They have a component for each node, which may overlap.

5 Conclusion

We proposed, implemented, and experimentally assessed a family of polynomial algorithms to compute the connected components of stream graphs. These algorithms handle streams of dozens of millions of events, and output connected components in a streaming fashion. This brings valuable information in practice, as we illustrate on a large-scale real-world dataset. We also propose a dataset approximation scheme making computations much faster while preserving key properties of the original data. Up to our knowledge, it is the first time that a partition of temporal nodes into connected components is computed at such scales.

A promising perspective is to enumerate connected components without listing them: one may for instance output only component size and duration in this way. Fully dynamic algorithms are particularly appealing to this regard, as their complexity is dominated by the explicit component listing.

References

1. Akrida, E.C., Spirakis, P.G.: On verifying and maintaining connectivity of interval temporal networks. *Parallel Process. Lett.* **29**, 1950009 (2019)
2. Alberts, D., Cattaneo, G., Italiano, G.F.: An empirical study of dynamic graph algorithms. *Exp, Algorithmics* **2**, 5-es (1997)
3. Bhadra, S., Ferreira, A.: Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks. *ADHOC-NOW* (2003)
4. Domenico, M.D., Lima, A., Mougél, P., Musolesi, M.: The anatomy of a scientific rumor. *Sci. Rep.* **3**, 2980 (2013)
5. Fontugne, R., Borgnat, P., Abry, P., Fukuda, K.: MAWILab: combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking (2010)
6. Fournet, J., Barrat, A.: Contact patterns among high school students. *PLoS One* **9**, e107878 (2014)
7. Gómez-Calzado, C., Casteigts, A., Lafuente, A., Larrea, M.: A connectivity model for agreement in dynamic systems. In: *Euro-Par 2015: Parallel Processing* (2015)
8. GroupLens Research: MovieLens data sets (2006)
9. Henzinger, M.R., King, V.: Randomized fully dynamic graph algorithms with poly-logarithmic time per operation. *ACM* (1999)
10. Huang, S.E., Huang, D., Kopelowitz, T., Pettie, S.: Fully dynamic connectivity in $O(\log n(\log \log n)^2)$ amortized expected time. *ACM-SIAM* (2017)
11. Huyghues-Despointes, C., Bui-Xuan, B.M., Magnien, C.: Forte delta-connexité dans les flots de liens. *ALGOTEL* (2016)

12. Isella, L., Stehlé, J., Barrat, A., Cattuto, C., Pinton, J.F., den Broeck, W.V.: What's in a crowd? Analysis of face-to-face behavioral networks. *JTB* **271**, 166–180 (2011)
13. Iyer, R., Karger, D., Rahul, H., Thorup, M.: An experimental study of polylogarithmic, fully dynamic, connectivity algorithms. *Exp. Algorithmics* **6**, 4-es (2001)
14. Kapron, B.M., King, V., Mountjoy, B.: Dynamic Graph Connectivity in Polylogarithmic Worst Case Time. In: *SODA* (2013)
15. Kejlberg-Rasmussen, C., Kopelowitz, T., Pettie, S., Thorup, M.: Faster worst case deterministic dynamic connectivity. In: *ESA* (2016)
16. Kempe, D., Kleinberg, J., Kumar, A.: Connectivity and inference problems for temporal networks. *JCSS* **64**, 820–842 (2002)
17. Kunegis, J.: Konect: the koblenz network collection. In: *Proceedings of the 22nd International Conference on World Wide Web*, pp. 1343–1350 (2013)
18. Latapy, M., Viard, T., Magnien, C.: Stream graphs and link streams for the modeling of interactions over time. *SNAM* **8**, 61 (2018)
19. Leskovec, J., Krevl, A.: SNAP datasets: stanford large network dataset collection (2014)
20. Mislove, A.: Online social networks: measurement, analysis, and applications to distributed information systems. Ph.D. thesis (2009)
21. Nicosia, V., Tang, J., Musolesi, M., Russo, G., Mascolo, C., Latora, V.: Components in time-varying graphs. *Chaos* (2012)
22. Paranjape, A., Benson, A.R., Leskovec, J.: Motifs in temporal networks. In: *WSDM* (2017)
23. Rannou, L.: Straph – Python library for the modelisation and analysis of stream graphs (2020). <https://github.com/StraphX/Straph>
24. Vernet, M., Pigne, Y., Sanlaville, E.: A study of connectivity on dynamic graphs: computing persistent connected components (2020)
25. Viswanath, B., Mislove, A., Cha, M., Gummadi, K.P.: On the evolution of user interaction in facebook. In: *SIGCOMM-WOSN* (2009)
26. Wulff-Nilsen, C.: Faster deterministic fully-dynamic graph connectivity. In: *SODA* (2013)
27. Xuan, B.B., Ferreira, A., Jarry, A.: Computing shortest, fastest, and foremost journeys in dynamic networks. *IJFCS* **14**, 267–285 (2003)



The Effect of Cryptocurrency Price on a Blockchain-Based Social Network

Cheick Tidiane Ba, Matteo Zignani^(✉), Sabrina Gaito, and Gian Paolo Rossi

Dipartimento di Informatica, Università degli Studi di Milano,
Via Celoria 18, Milan, Italy

{cheick.ba,matteo.zignani,sabrina.gaito,gianpaolo.rossi}@unimi.it

Abstract. The massive spread of online social media platforms has favored the emergence of social media giants, like Facebook, Twitter, Instagram. These companies have been the center of many scandals related to privacy issues, data ownership and censorship. The issues that stem from their centralized architecture, brought the spotlight on alternative solutions based on decentralized or distributed architectures. In these alternative social media platforms, the data is not owned by a single company, the open nature makes censorship harder and users can profit from their contents. Here we focus on a specific solution - Steemit - built on top of a public blockchain, linked to the Steem cryptocurrency. In fact, in Steemit, the participation and the content quality are rewarded with a cryptocurrency through a network- and user-based voting system. This way, the network structure, the dynamics on top of it and the cryptocurrency market are strongly coupled. Such an interplay is the focus of the paper; specifically, we study the impact of the cryptocurrency Steem on the social network growth, using more than 4 years of data extracted directly from the Steem Blockchain. We find that the growth of the network is strongly tied to the fluctuations in Steem cryptocurrency price: it can be observed that rising Steem prices trigger the network growth, and similarly, when Steem value drops, so does the network's growth. We also find evidence of a lead-follow relationship between Steem price and users' behavior: our study suggests that the full impact of the Steem cryptocurrency price can be observed within 3–4 weeks. So essentially, the blockchain-based social network Steemit represents a valuable case study among the emerging online social media, where network dynamics and economic/financial aspects are strongly intertwined; and where the underlying blockchain is an unprecedented source of data for measuring such interplay.

Keywords: Network evolution · Blockchain-based social network · Cryptocurrency

1 Introduction

In the last decade, we have witnessed the massive spread of social media platforms such as Facebook, Twitter, Instagram. These platforms are usually controlled by one social media company, that owns all the data and decides its own

policies. This propensity towards centralization also reflects on their platform architecture, where a single server, hold and managed by the company, is in charge of all the services and data¹, leading to what is known as centralized social networks. As online social networks (OSNs) have become widely used, many issues related to the centralized approach have emerged. The first concern regards data ownership and data monetization, and is strictly related to the business model they adopt. In fact, targeted advertising based on user data is one of their main revenue sources, but generally users do not receive any rewards from the data we provide, even tough content is what keeps other users engaged. Second, the centralized approach opens up several privacy issues, not only related to a single point of access to data breaches, but also to the improper usage of private data. Finally, the same centralized approach together with a full data ownership has consequences on the censorship, since the company can censor any content on the platform based on self-imposed guidelines or on political pressure.

These above issues have questioned the centralized approach and spotlight different alternative architectures such as decentralized or distributed social networks. Even though the eco-system of these platforms is quite varying, generally, data is not owned by a single company but is stored on independent servers or through distributed technologies like peer to peer networks, or blockchains. Moreover, most of these social networks offer data policies which give the content ownership back to the users. Among these alternative social networks, blockchain-based OSNs - Steemit, Sapien or Minds - are the most interesting as they are usually tied to cryptocurrencies that can be spent for goods and services or gained by the creation of high-quality contents. But, this is not an original aspect, since, even in mainstream platforms users are rewarded for content creation through the ads-system. The novelty is the lack of any content advertisement or content recommendation system, and the presence of a reward distribution mechanism which is: *i)* based on a voting system where both creators and curators may get revenues in producing/promoting high-quality contents; *ii)* strongly tie with a unstable cryptocurrency market; and *iii)* well-known among the platform users and not modifiable by the platform administrators. In this scenario, where the reward system, the cryptocurrency market and the network structure are tightly intertwined, we look to answer at the following question: in online social networks that rely on cryptocurrency, what is the impact of the cryptocurrency price on the network growth and on social actions like upvoting or sharing?

To answer this question, here we focus on Steemit, one of the most popular blockchain-based social networks with more than 1 million users. In Steemit users make blog posts with content, that can be shared and voted by other users, and creators and curators posts are rewarded with Steem cryptocurrency. Since every action is recorded on the public Steem blockchain, we were able to collect a dataset that describes more than 4 years of social network activities, covering more than 130 millions of follow relationships. Each event has its own

¹ Actually, this is an abstraction of the architecture, since modern online social networks heavily rely on large-scale data center and content delivery networks.

timestamp, so we can study the impact of the price of the cryptocurrency Steem² on the social network growth. From the analysis based on temporal correlation, we noticed that the growth of the network is strongly tied to the fluctuations in Steem cryptocurrency price: it can be observed that rising Steem prices trigger the link creation process and similarly when Steem value drops, so does the network's growth. In fact, there is a positive correlation (0.7) between Steem price values and the creation of "follow" relationships. By the analysis on correlation lags, we also found evidence of a lead-follow relationship between time series, where Steem prices influence user behavior: our cross-correlations study suggests that the full impact is felt with a 25–32 days difference. Finally, by analyzing where links form during these correlation phases, we identified suspicious behaviors in high in-degree nodes which try to favor the diffusion of particular contents, so doping the rewarding system. In general, we have highlighted that, in blockchain-based social networks which implement reward mechanisms relying on cryptocurrencies, the network dynamics and economic/financial aspects are strongly intertwined, and in the former, growth patterns not explainable through social theories come into play.

The rest of the paper is structured as follows. In Sect. 2, we introduce the blockchain-based social network Steemit and a brief review about studies on Steemit. Then, in Sect. 3 we describe how we collect data from the Steem blockchain. In Sect. 4 we describe the methodology to assess the temporal correlation among the network dynamics and the price of the Steem cryptocurrency. Finally, in Sect. 5 we discuss our main findings about the temporal correlations and the identification of suspicious behaviors involving high in-degree accounts.

2 Steemit: A Blockchain-Based Online Social Network

In this section we summarized the main characteristics of the components making up the Steemit social network. Steemit is one of the most popular blockchain-based OSN with more than 1 million users. In Steemit users make blog posts with content that can be shared³ and voted by other users. According to these functionalities, a user can be a *creator* - content producer - or a *curator* - content promoter. The most popular posts rise in visibility, and post creators and curators of the top posts are paid with the Steem cryptocurrency. The Steemit platform relies on the Steem-blockchain [3] for data storage and action tracking: every operation is stored on the blockchain, making the platform more resistant to censorship - all changes are persistent. The data is not owned by the Steem company, since the blockchain is publicly available by entering in the P2P network or through API. Finally, as the system is sustained by selling tokens, users' data are not exploited by the Steemit company to support targeted advertisements.

² Cryptocurrency price history data is recovered from an already existent web service.

³ Resteemed, in the Steemit jargon.

Steem-Blockchain. A blockchain is a list of records, called blocks, characterized by a timestamp and data. Blocks are linked through cryptography and each block has a hash of a previous block. This concept was made famous by Bitcoin [16]. With a blockchain, records are stored publicly and distributed to all the servers inside the P2P network. Every new transaction must be verified by the users in the network. The verification is regulated by a consensus protocol: the transaction is encrypted and sent to all users, if the transaction is considered valid by the majority of users, a new block is created with the transaction and sent to all users. In traditional blockchain we have Proof of Work (PoW). In PoW, users - miners - compete to solve a complex mathematical problem to verify a block. The first miner to complete the task creates a new block for the blockchain and is rewarded for its effort. The Steem-Blockchain started with PoW but then moved on to a variant, called Delegated Proof of Stake (DPoS) [12], to speed up the verification process and handle high-frequency events typical of online social networks. In DPoS, we do not have miners; instead block production is assigned to a subset of users, called witnesses. Every user can be elected witness: users can trade Steem cryptocurrency into Steem stakes to give their vote more power. Witnesses are incentivized as there is a reward for block producers as well. Block production is done in rounds, and the witnesses are rotated each round. This approach reduces costs as mining rigs are not necessary and it allows to produce a block every 3 s.

Steem Cryptocurrency. Steem's cryptocurrency system is composed by three different types of currency. The main currency issued by Steem is Steem which is used for trading; this unit has an actual value in terms of real money⁴ and can be acquired and sold: for example, one can use Steem on various exchanges to convert it to Bitcoins, to other cryptocurrencies or to traditional currencies. The other two currencies are Steem Dollars and Steem Power. They are dependent on Steem and they are the main forms of payment for content creators and curators. Steem Dollars are a stable currency, where 1 Steem dollar represents the amount of Steem required to reach 1 US Dollar (USD). However, according to the Steem FAQ, it could be worth more or less than 1 USD depending on market conditions expressed by the exchange rate. So, Steem Dollars can be spent on goods or traded for other currencies. The Steem Power is the equivalent of market shares in Steem: just like real life shares, if the value of the company increases, so does the value of the user shares. By investing Steem Dollars or Steem the currency is turned in Steem Power. Users' Steem Power capital is fundamental in the distribution of the reward since its associated voting power decides the share of revenue for curators that upvoted the most popular posts. Moreover, those who invested in Steem Power use their stakes for voting both for posts and for witness election.

Studies on Steem. Although research field about blockchain-based solutions and networks resulting from cryptocurrency transactions is very active

⁴ The actual Steem value is available at <https://coinmarketcap.com/currencies/steem/>, for example.

([6,9,14,15] to cite a few studies), blockchain-based social networks and their specific characteristics are not fully understood yet. Only recently Guidi [7] has published an extensive survey on decentralized and distributed online social network which also covers blockchain-based OSN main features, open problems and possible solutions. However, since the release of the seminal white paper on the platform [3], only a few works have focused on Steemit. For instance, Chonan [4] and Kim *et al.* [11] have focused on the structure of its social network and its characteristics from a static viewpoint, only. An analysis oriented towards the diffusion of contents at a mesoscopic scale has been conducted by Jia *et al.* [10]. They have studied the distribution of votes and comments around 5 popular tags and their related subgroups, but they have not coped with dynamical or financial aspects. The latter aspect have been deepened in Li *et al.* [13], where they have described and analyzed the networked structures behind the Steemit rewarding system. Specifically, they have focused on the rewards misuse and bot abuse in Steemit with Steem-blockchain data up to August, 2018. They have also found some visual evidences of correlation between changes in Steem price and the monthly increment of users and operations; but they did not measure any effect on the growth of the network.

3 Dataset

In order to study the interplay between the social and financial aspects in Steemit we needed the longitudinal data of the cryptocurrencies that influence the Steem value over time. We can retrieve such data from [2]. There we can consult the daily value of the Steem currency in US Dollars and other cryptocurrencies. The prices are updated daily and we find values from April 18th, 2016. So, from this platform we collected data for the Steem price in USD. Alongside the Steem value, we also consider Bitcoin, the leading cryptocurrency, as it is shown that it has strong influence on the value of other cryptocurrencies.

Then, as our goal is studying the currency's impact on the evolution of the Steemit social network, we had to recover data describing the relationships among users. The collection of these operations composes a detailed temporal evolution dataset, that describes user activity with a temporal precision of 3 s⁵. In Steemit every operation can be retrieved from the Steem blockchain: researchers and application developers have access to the data through a series of APIs, that can be queried through HTTP requests. Relying on the Steem-python library, we were able to specify a Steem node (<https://steemdev.api.com>) to handle our data requests. We collected the data from the very first block, produced on 24th March, 2016, up to block 44301097, that was produced on 16th June, 2020 - an overall period of 4 years.

As introduced in Sect. 2, users on Steemit can perform many different actions. According to the official documentation [1], on the blockchain we can find more than 50 different types of operation. As we are interested in the relationships in the social network, we had to extract “follow” information. This data is stored

⁵ The action timestamp is derived from its block and each block is verified every 3 s.

in `custom_json` operations. In the “`json`” field users can input any kind of data as long as it follows the JSON format: thus requiring a filtering step to actually extract “follow” relationships. According to the documentation, we can filter “follow” operations by looking at the presence of the field `id`, set to `follow`. In our collection, we obtain a total of 157,883,036 “follow” operations with their timestamp. “Follow” actions cannot be directly used to build the social graph, since Steemit tracks more than just follow operations. For example, users can decide to perform an “unfollow” action. Not only, Steemit also allows to completely block a user. Alongside these three main actions, some of the collected operations may be not properly formatted: usually they are the result of errors made by developers or user-made scripts. Therefore we must further filter “follow” operations, and, according to the official documentation, group them based on the `what` attribute. We focus on three main options: `blog`, which is the equivalent of a real following action; `ignore` which is the equivalent of a blocking action, and, finally, we consider empty strings as unfollowing actions. After filtering, we obtain 134,941,606 “follow” relationship, 20,216,913 “unfollows” and 2,721,355, “ignore/mute” (empty strings or arrays, or explicit “muted”) actions. It is worth to note that the “follow” actions were not immediately available in the platform from the beginning, but the introduction of this functionality dates back to June 3rd, 2016 (commit e8472fb), according to Steem project commit history⁶.

4 Methods

As mentioned, our objective is studying the relationship between Steem price values and users’ social behavior. In our analysis, we focus on the daily evolution of the network. We aggregate “follow” operations by date, obtaining a time series describing the number of new “follow” relationships in the social network, every day. Thus, we have obtained two time series: the daily new “follow” links and the historical data of the daily Steem price.

We first look at potential seasonal patterns through the analysis of the *Auto-correlation Function* (ACF). It measures the linear relationship between lagged values of a time series; the resulting plot shows if data has some sort of pattern, either a long-term trend or a seasonal pattern [8]. The ACF is the function of autocorrelation values ρ_k for every lag k , where $\rho_y(k)$ is defined as

$$\rho_y(k) = \frac{\sum_{t=k+1}^T (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^T (y_t - \bar{y})^2}$$

According to [8], if data are trended, $\rho_y(k)$ values for small lags are large and positive; observations closed in time are also close in size. So, the ACF of trended time series will show positive values that slowly decrease as the lags increase.

⁶ <https://github.com/steemit/steem/search?o=asc&q=follow&s=committer-date&type=Commits>.

When data are seasonal, the autocorrelations $\rho_y(k)$ will be larger for the seasonal lags (at multiples of the seasonal frequency) than for other lags. When data are both trended and seasonal, we can observe both these phenomena.

After studying the singular time series, we shift our focus on the relationship between the two time series. We compare the time plots for the series looking for potential visual evidence of influences or correlations; then we use scatter plots and quantify the correlation between two time series using correlation coefficient. We evaluate the correlation by the classical *Pearson Coefficient* [5]. Given two time series X and y , we compute the *Pearson Coefficient* $\rho(x, y)$ as:

$$\rho(x, y) = \frac{\sum(x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum(x_t - \bar{x})^2} \sqrt{\sum(y_t - \bar{y})^2}}. \tag{1}$$

with values towards 1 indicating perfect correlation, 0 no cross-correlation and around -1 perfect anti-correlation. This measure tell us if there is a linear relationship between the Steem price and the amount of “follow” actions.

Finally, we can determinate potential lead-follow relationships between time series using the *normalized cross correlation* measure. Given two time series x and y , the normalized cross-correlation measure is similar to the correlation measure: instead of correlating once x with y , we do it multiple times, considering the time series y , but shifted by a series of time lags k . We obtain a series of different correlation values ρ , one for each chosen time lag k . In our work, we consider lags in days. This measure can be expressed as:

$$\rho_{xy}(k) = \frac{\sigma_{xy}(k)}{\sigma_x \sigma_y} = \frac{\sum_{t=k+1}^T (x_t - \bar{x})(y_{t-k} - \bar{y})}{\sum_{t=1}^T (x_t - \bar{x})(y_t - \bar{y})} \tag{2}$$

This process produces a set of pairs (lag, correlation value). We can better explore them by analyzing their shape and focusing on the time lags k that show the highest correlation values. If we find high correlation values for a positive time lag, then x leads y ; vice versa, if the higher values are for a negative time lag, then we have that time series y is leading x .

5 Results

In this section we present the insights obtained by the methodology introduced in the previous section. Specifically, in Fig. 1 we report the auto-correlation function for the two time series: a) new “follow” relationships and b) Steem price, on daily basis.

As shown in Fig. 1a, the auto-correlation function on the new “follow” time series shows the lack of repeating peaks, which means there are no seasonal trends. Also, we can observe that the new “follow” is a trended time series, since it tends to have positive values that slowly decrease as the lags increase. We obtain a similar result while evaluating the Steem price time series (see Fig. 1b), suggesting the lack of seasonal trends.

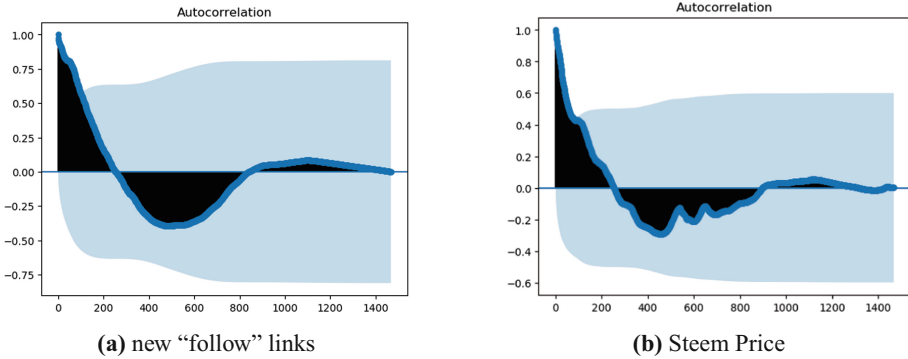


Fig. 1. The auto-correlation function for the *a)* new “follow links and *a)* Steem price. On the Y-axis: the correlation coefficient $\rho_y(k)$. On the X-axis: the lag k in days.

The Interplay Between Network Growth and Steem Price. Before delving into a quantitative assessment of the temporal correlation between the network evolution process and the Steem price, we qualitatively inspected the trends of the two time series, to verify if our hypothesis of correlation holds. In fact, it was also observed in [13], that the activities, such as comments, votes, content sharing and node arrival, could be strongly tied to the fluctuations in Steem cryptocurrency price, but the observations hold for a shorter period w.r.t. our dataset. In fact, looking at Fig. 2, we can confirm those observations for the periods of growth: April 2017 - June 2017, November 2017 - January 2018 (cyan regions in the Fig. 2).

Analyzing the new dataset, we can also observe a similar influence in successive periods of time. Around March, 2018 we see that as the price of Steem falls, so does the activity in terms of “follow” operations. Around the beginning of April 2018, we can see a small rise in Steem price; it precedes one of the biggest growth of the “follow” relationships in the network. This spike is however short lived, as the Steem price falls again. We can see that shortly after the number of “follow” operations per day starts to shrink: we reach the lowest level recorded up to that point. The Steem price has never recovered and is now hovering around 0.20 USD. This is an important new phenomena: not only the success of the cryptocurrency is an important catalyst of a social network’s growth, but we also saw that a drop in value has stunted the growth of the network. The crisis that emerges from the data was also confirmed by the Steemit company: a post by at the time Founder and CEO of Steemit, Inc., Ned Scott, on 28/11/2018, confirms the crisis: Steemit had to lay down 70% of its workforce as the maintenance costs were becoming too high [17].

Given, the previous evidences, we evaluated the strength of the correlation between the Steem cryptocurrency and the number of “follow” relationships that are created daily in Steemit. We first compute the Pearson correlation coefficient (Eq. 1): we obtain an important positive correlation of 0.71, that confirms that

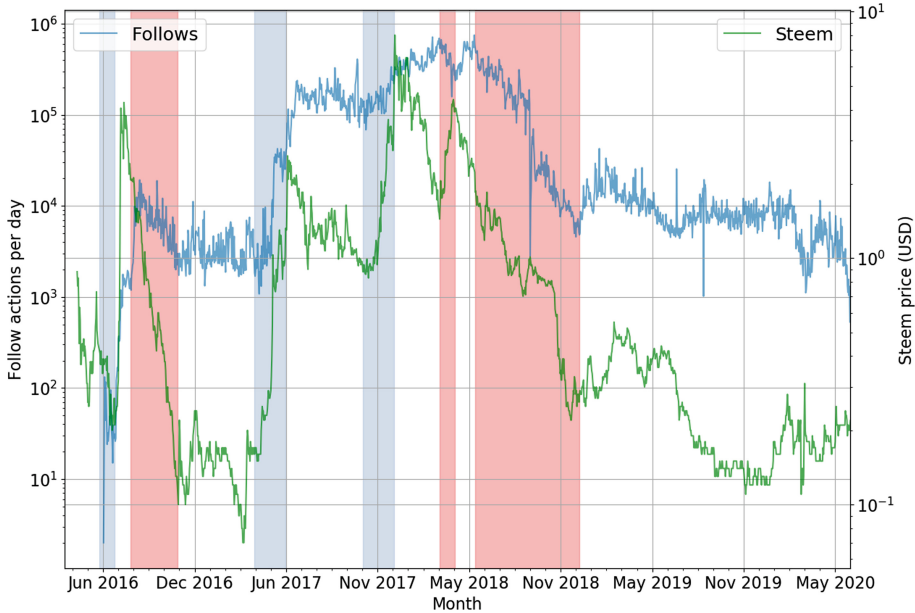


Fig. 2. Time plots for “follow” operations (blue) and Steem price value in USD (green). On x-axis: time in months, from June 3rd, 2016 (“follow” plugin in Steemit) up to June 3rd, 2020. On left Y-axis: volume of “follow” operations per day. On right Y-axis: Steem price in USD.

Steem changes and users behavior are strongly connected. In our hypothesis, Steem prices are influencing user activity, and considering we are not taking into consideration the time it would take for users to react to Steem fluctuations, this is a pretty high correlation value. The corresponding scatter plot is displayed in Fig. 3a and shows that low activity days tend to be linked to low Steem price and higher activity days tend to be linked with higher values of the cryptocurrency.

Finally we are able to determinate potential lead-follow relationships between two time series using the *normalized cross correlation* measure between the Steem price and the “follow” relationships. We compute different cross correlation values by Eq. 2. We test lags in the range of $(-90, +90)$, and we obtain 180 correlation values for these time lags. In Fig. 3b we display the plot of the pairs of lags and correlation values. In the figure, we look for the range of days that register the highest correlation values. We obtained positive moderate cross-correlation values across the whole interval (>0.5). The highest value is 0.87, obtained considering a lag of 32 days. This high correlation value confirms that, indeed, there is a lead-follow relationships between the two time series. These lags suggest that the full impact of Steem price on the network evolution is felt with a 25–32 days difference.

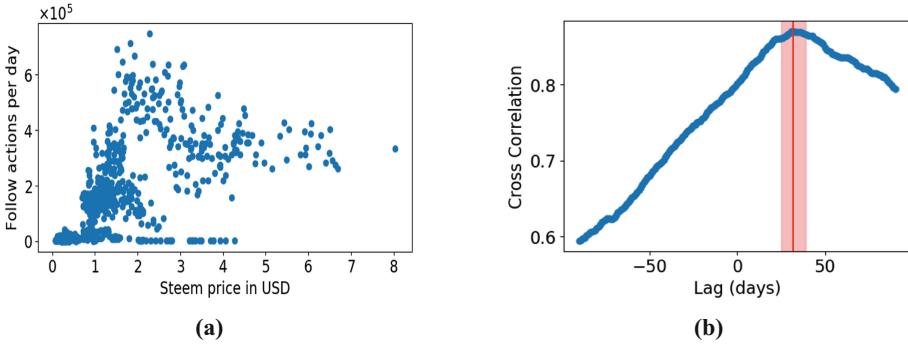


Fig. 3. In *a*) The scatter plot of Steem price and “follow” relationships per day. In *b*) the normalized cross correlation between the Steem price and the daily number of “follow” relationships. On the X-axis the lag and on the Y-axis the coefficient value.

Active Users and Bot Accounts. In Sect. 5 we have discussed the presence of two periods of growth: April, 2017 - June, 2017 and November, 2017 - January, 2018. We want to understand which users are the most active in the two periods. Starting from the “follow” actions we isolate those occurring in the two above periods. This allows us to create two subgraphs representing the network’s evolution in those intervals. Taking the nodes with highest in-degree, we can pinpoint the most popular users. This way, we find that among the most popular accounts, we see that some of them have “reesteem” or “bot” in their names. In Steemit, a reesteem is the equivalent of sharing a content/post. Sharing is an important activity: when a user shares a post, it is going to be shared to all of his followers. A user is incentivized to share posts he voted or commented, as only popular posts will be rewarded with currency. So, sharing a post is just a further investment. We decided to investigate these profiles. One of them even clearly states that the accounts will automatically share every post from his followers, to all the followers. Observing the actions, we can see that indeed the main actions shown in their history are “reesteem” actions and seems to be done automatically. These bots provide an advantage to all the users: users that use the bot to gain visibility for their posts, increasing the chance of their post becoming popular and the rewards. At the same time, the bot has the opportunity to gain from the curation (vote, comment, sharing) of posts from the users. Sharing a post makes it more likely it gains popularity, increasing the chance of a reward for both the author and the bot.

6 Conclusions

In this paper, we have studied Steemit, one of the leading blockchain-based decentralized social network. In these social networks, creators and curators are rewarded with cryptocurrency for their efforts. Our objective was to study the relationship between the cryptocurrency and the growth of the social network.

We did so by analyzing more than 4 years of daily data for the users' social activity and the price value of the Steem cryptocurrency. The analysis shows that the growth of the network is strongly tied to the fluctuations in Steem cryptocurrency price: we observed that rising Steem prices trigger network growth and when the Steem value drops, so does the network's growth. A correlation analysis confirms that there is a strong positive correlation (0.7) between Steem price values and network activity. We also confirmed that there is lead-follow relationship between time series, where Steem prices influence user behavior. The studied lags suggest that the full impact is felt with a 25–32 days lag. In conclusion, we show that the cryptocurrency rewards influence social network growth and user behavior, for better or worse. This work shows that while a cryptocurrency reward can be a strong incentive to join a blockchain based network, it can also be a social network's downfall.

References

1. Broadcast Ops. <https://developers.steem.io/apidefinitions/broadcast-ops>
2. Steem (STEEM) price, charts, market cap, and other metrics. <https://coinmarketcap.com/currencies/steem/>
3. Steemit Whitepaper. <https://steem.com/steem-whitepaper.pdf>
4. Chohan, U.W.: The concept and criticisms of steemit. Available at SSRN 3129410 (2018)
5. Freedman, D., Pisani, R., Purves, R.: Statistics (International Student Edition), 4th edn. WW Norton & Company, New York (2007). Pisani, R. Purves
6. Gensollen, N., Latapy, M.: Do you trade with your friends or become friends with your trading partners? A case study in the g1 cryptocurrency. *Appl. Netw. Sci.* **5**(1), NA–NA (2020)
7. Guidi, B.: When blockchain meets online social networks. *Perv. Mob. Comput.* **62**, 101131 (2020)
8. Hyndman, R., Athanasopoulos, G.: Forecasting: Principles and Practice. OTexts, 3rd edn. (2019). <https://Otexts.com/fpp3/>
9. Ji, Q., Bouri, E., Gupta, R., Roubaud, D.: Network causality structures among bitcoin and other financial assets: a directed acyclic graph approach. *Q. Rev. Econ. Financ.* **70**, 203–213 (2018)
10. Jia, P., Yin, C.: Research on the characteristics of community network information transmission in blockchain environment. In: 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), vol. 1, pp. 2296–2300 (2019)
11. Kim, M.S., Chung, J.Y.: Sustainable growth and token economy design: the case of steemit. *Sustainability* **11**(1), 167 (2019)
12. Larimer, D.: DPOS consensus algorithm—the missing whitepaper, steemit (2018)
13. Li, C., Palanisamy, B.: Incentivized blockchain-based social media platforms: a case study of steemit. In: Proceedings of the 10th ACM Conference on Web Science, pp. 145–154 (2019)
14. Maesa, D.D.F., Marino, A., Ricci, L.: Data-driven analysis of bitcoin properties: exploiting the users graph. *Int. J. Data Sci. Anal.* **6**(1), 63–80 (2018)
15. Maesa, D.D.F., Marino, A., Ricci, L.: The bow tie structure of the bitcoin users graph. *Appl. Netw. Sci.* **4**(1), 56 (2019)

16. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system. Cryptography Mailing, March 2009. <https://metzdowd.com>
17. Scott, N.: Steemit update, November 2018. <https://steemit.com/steem/@ned/2fajh9-steemit-update>



Multivariate Information in Random Boolean Networks

Sebastián Orellana^(✉) and Andrés Moreira

Departamento de Informática, Universidad Técnica Federico Santa María,
Valparaíso, Chile

sebastian.orellan.12@sansano.usm.cl, amoreira@inf.utfsm.cl

Abstract. In the study of complex networks, simple local heterogeneous interactions favor highly complicated and non-linear dynamics. In this paper, we take advantage of recent advances presented by Rosas et al [Physical Review E, 100, 032305] to capture the fundamentals of dynamics: high-order interdependencies. In particular, the phase diagram of Random Boolean Networks is described in terms of the information shared between multiple nodes. We found that the critical point between ordered and chaotic regimes is well defined by a balance between redundancy and synergy, for both normal and scale-free topologies. In addition, particular network structures are identified that characterize the behavior of high-order interdependencies in each dynamic regime.

Keywords: O-information · Phase diagram · Random boolean networks

1 Introduction

Over the last decades several approaches have tried to investigate the dynamics that govern complex systems, in order to understand how they process information. This old question will remain largely open as long as we fail to grasp the fundamentals of the dynamics: the interdependencies involving a large number of agents, wherein the richness of the system lies, rather than in the agents' features. Recently, building on previous work [2, 7, 20, 21] perfected *O-information*, a promising generalization of *mutual information*, as an effective metric to quantify statistical high-order interdependencies among the agents of a system. It distinguishes between the synergistic or redundant nature of such interdependencies, whose dynamic relevance has been widely demonstrated [6, 21].

These advances from information theory may be applied to the analysis and design of dynamical systems. Of particular interest is the study of complex network dynamics, where heterogeneous interactions involve multiple influences between components, favoring complicated and highly non-linear behaviors, and the dynamics is hard to predict from local interactions. In this context, a natural candidate are Random Boolean Networks (RBNs), which correspond to a

type of dynamic network capable of reproducing very complex behaviors, while maintaining a framework simple enough to aspire to theoretical results.

An interesting property of RBNs is that they present a well-known transition between order and chaos with respect to their average connectivity \bar{K} and the bias of their Boolean functions. It has been argued that this critical point maximizes dynamic complexity and optimizes information processing [4, 11]. Intuitively, it is efficient to live on the edge of chaos to benefit both from the easy spread of information of a chaotic regime, as well as the ability to store such information, characteristic of an orderly regime.

Along this line, [14, 18] describes the optimal information processing regime in RBNs based on the *mutual average information* (AMI) on all pairs of nodes, as a measure of how well coordinated the internal dynamics is, finding in fact a maximization of this metric near the critical point. Similarly, [13] employs the *information dynamics* to explain the transition phase in terms of the maximization of information storage and coherent information transfer. However, neither approach uses an appropriate multivariate generalization of *mutual information*, and are therefore insensitive to high-order interdependencies between the nodes of the network (those involving three or more nodes).

Thus, following the spirit of [14, 18] and [13], this work seeks to reinforce the bridge between complex networks and information theory, by describing the phase diagram of RBNs in terms of the high-order interdependencies that govern their dynamics. In addition, we explore the structural features that favor each regime. The article is organized as follows: Sect. 2 introduces the preliminary concepts for calculating *O-information* in RBNs, Sects. 3 and 4 examine a description of the phase diagram of RBNs as a function of high-order interdependencies, and finally Sect. 5 presents the relevant conclusions.

2 Preliminaries

High-Order Interdependencies in Boolean Networks. Consider a Boolean network of n nodes, and denote the set of states (or a configuration) at time t as $\mathbf{X}^n(t) = (X_1(t), \dots, X_n(t))$, where $X_i(t) \in \{0, 1\}$. Then the state of the i -th node at time $t + 1$ is determined by $X_i(t + 1) = f_i(\mathbf{X}^n(t))$, where the Boolean function f_i includes connectivity, that is, it depends on $X_j(t)$ if and only if there is an edge from node j to i .

An orbit is defined as the sequence of configurations starting from a given initial condition. With this, the dynamics of the network can be described as an ensemble of orbits produced from different initial conditions. It can also be seen as an ensemble $\mathbf{X}^n = (X_1, \dots, X_n)$ of individual trajectories, where X_i is a binary vector describing the states of node i along the orbit. Note that a probability distribution over the 2^n possible initial conditions induces a distribution for \mathbf{X}^n . The ensembles of trajectories can be build in several ways, depending on the duration of the orbit, the sampling of the initial configurations, and possible restrictions to orbits in attractors of the system; this will be specified in the different sections.

On \mathbf{X}^n it is feasible to apply the framework introduced in [7] and recently explored in [20] for the calculation of high-order statistical interdependencies in a dynamic system. For this, first consider the *binding information*:

$$B(\mathbf{X}^n) = H(\mathbf{X}^n) - \sum_{i=1}^n H(X_i|X^{-i}), \tag{1}$$

which describes the average shared information between two or more nodes of the network, with $X^{-i} = (X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n)$. Note that $H(X_i|X^{-i})$ corresponds to the residual information $R(X_i)$ of the node i , that is, all information that can only be retrieved by directly accessing the dynamic information of i . Then consider the *total correlation*:

$$C(\mathbf{X}^n) = N(\mathbf{X}^n) - \sum_{i=1}^n N(X_i) = \sum_{i=1}^n H(X_i) - H(\mathbf{X}^n), \tag{2}$$

which recovers the average amount of collective restrictions imposed by the *negentropy* $N(\mathbf{X}^n)$ [2]. Note that $H(X_i)$ corresponds to the individual entropy of node i . Finally the *O-information* is defined as:

$$\Omega(\mathbf{X}^n) = C(\mathbf{X}^n) - B(\mathbf{X}^n) \tag{3}$$

In addition to being able to capture the magnitude of interdependencies involving three or more agents within a system (for more detail see [20]), it also discriminates the prevalent form of high-order interdependencies. In particular, if $\Omega(\mathbf{X}^n) > 0$ the system is said to be dominated by redundant interdependencies, while if $\Omega(\mathbf{X}^n) < 0$ the system is said to be dominated by synergistic interdependencies. As $\Omega(\mathbf{X}^n)$ approaches its limits ($2 - n \leq \Omega(\mathbf{X}^n) \leq n - 2$ for the Boolean case) the dominance is accentuated, while values $\Omega(\mathbf{X}^n) \approx 0$ define a synergistic-redundant balance. We will therefore refer to the *binding information* and the *total correlation* as the synergistic and redundant components of the *O-information*, respectively.

Random Boolean Networks (RBN). RBNs, originally proposed as models of genetic regulation networks [8], offer an ideal theoretical framework for the statistical study of interdependencies in Boolean networks. Kauffman’s original model generates Boolean networks by randomly assigning to each node K inputs and an update function.

An interesting feature of RBNs is that they present a well-known transition point between order and chaos depending on their average connectivity \bar{K} . In [3] it was analytically determined that the critical point of the RBN¹ is found when:

$$\bar{K}_c = 1/2p(1 - p), \tag{4}$$

¹ It should be noted that this result is of a statistical nature, since it is possible to find chaotic networks in the predicted stable regime and vice versa.

where p corresponds to the probability that each assigned Boolean function produces a ‘1’ in its output. Furthermore, [1] produced an alternative expression of these results for RBNs with scale-free topologies, which is used in the next section to strengthen the study of high-order interdependencies in RBNs.

3 O-Information in Random Boolean Networks

An experiment was done with 500 RBNs of size $n = 20$ for different values of \bar{K} and p . Here, \mathbf{X}^n corresponds to the simulation of 50 iterations starting from the 2^n possible initial configurations. Then $\Omega(\mathbf{X}^n)$ is calculated for each case.

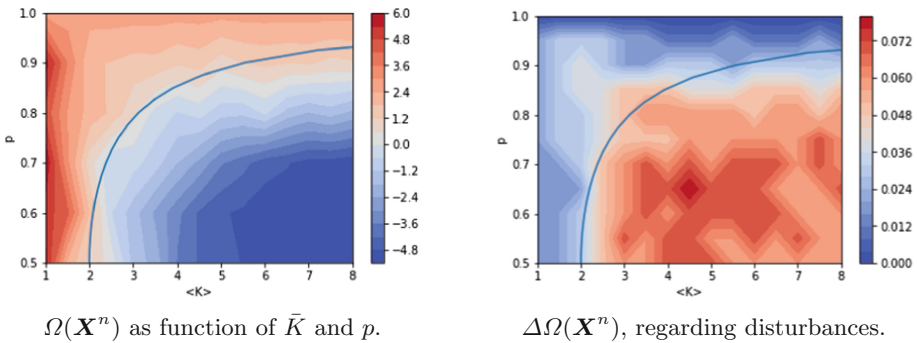


Fig. 1. High-order interdependencies as a function of \bar{K} and p , in RBNs of size $n = 20$ and a normal degree distribution. The blue curve corresponds to the theoretical critical point in RBN.

In order to contrast the results with the phase diagram of the RBNs presented in [3], Fig. 1 corresponds to a 2D projection of $\Omega(\mathbf{X}^n)$ as a function of \bar{K} and p . Here it is observed that the O-information is effectively capable of capturing relevant aspects of the dynamics, since it is clearly sensitive to the structural characteristics of the network. The most interesting finding is that the critical point in RBNs is defined by a balance between redundancy and synergy, corresponding to $\Omega(\mathbf{X}^n) \approx 0$. With this, as the chaos in the dynamics increases, the ability to store the same information in different nodes is lost, in exchange for the ability to share information in more complex ways spread throughout the network. Likewise, the tendency to order in the dynamics hinders this propagation of information in favor of robustness. In general, $\Omega(\mathbf{X}^n)$ is able to capture the phase diagram of the standard RBN model, attributing redundant characteristics to the ordered regime and synergistic characteristics to the chaotic regime. Thus, in the interval $p \in [0.5, 1]$, $\Omega(\mathbf{X}^n)$ exhibits a decreasing behavior with \bar{K} and increasing with p . This allows us to hypothesize that within both the ordered and chaotic phases there is a certain continuous change determined by the magnitude of $\Omega(\mathbf{X}^n)$. That is, the greater the distance to the critical point, the more ordered (or chaotic) the dynamics will be.

Next we examine how the quantification of high-order interdependencies is related to a network's sensitivity to perturbations. Figure 1b shows the average variation of $\Omega(\mathbf{X}^n)$ when considering pairs of initial configurations with Hamming distances equal to 1. Again, *O-information* captures the phase diagram of the RBNs, but this time regarding the robustness of each regime. Although the gradualness observed in Fig. 1a is lost as we move away from the critical point, it is clear that an orderly regime induces very robust high-order interdependencies, while in a chaotic regime, the amount of the information shared between the nodes in a network is more sensitive to the initial configuration.

The Importance of Network Topology. Kauffman's original RBNs [8] considered only random regular graphs (same K for all nodes), or alternatively a topology with normally distributed degrees, centered on the average \bar{K} . This implies the absence of nodes that significantly depart from the norm, or of any community structure, which are features of many networks, including real-world regulatory networks. In general, there is evidence that even for very simple functions the topology can be decisive for the dynamics [1, 4, 16]. A particularly studied case are networks with topologies whose degree distributions follow a power law, where a few nodes have a very high degree and many nodes have a low degree. Studies on RBNs with such scale-free topologies shows a favoring of the evolution and adaptation of the functioning of the network from a biological perspective [1], and in general the correlation between node pairs increases [16].

Thus, as the *O-information* is capable of revealing relevant dynamic aspects, it is suspected that it is sensitive to structural variants of the RBNs. Indeed, by replicating the analyzes with scale-free topologies ($P(K) \sim K^{-\gamma}$), again $\Omega(\mathbf{X}^n)$ can capture the phase diagram presented in [1] assigning order and chaos to redundant and synergistic information respectively (Fig. 2a). Furthermore, the optimal processing regime tends to correspond to the balance between both types of information shared between the network, although with significantly larger deviations than those presented in Fig. 1. Such discrepancies are likely due to the small size of the networks considered, which introduces wide variations for scale-free topologies. Even so, Fig. 2a relates to [1] by noting that the rate of greatest sensitivity of $\Omega(\mathbf{X}^n)$ with respect to p occurs in the interval $\gamma \in [2, 2.5]$ (also the interval where the critical point is found), all scenarios $\gamma > 2.2$ being completely dominated by redundant interdependencies. Regarding the perturbation analysis, Fig. 2b shows a behavior analogous to the standard case.

An interesting aspect is that in both topologies the statistical range of $\Omega(\mathbf{X}^n)$ tends to be much more limited than the theoretical range. With values ranging through a window centered on balance, a more heterogeneous distribution appears to offer a slight tilt towards synergistic dominance, while a homogeneous distribution slightly favors redundant dominance. In any case, the extremes of the *O-information* seem to be hard to reach for random structures, which is accentuated in scale-free networks, possibly as a result of the expansion of the optimal processing regime suggested in [1]. Furthermore, the components $B(\mathbf{X}^n)$ and $C(\mathbf{X}^n)$ tend to be higher in networks with scale free topologies (images not

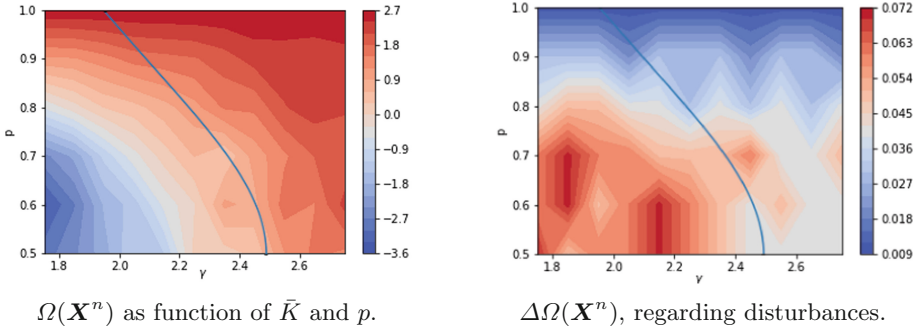


Fig. 2. High-order interdependencies as a function of \bar{K} and p , in RBNs of size $n = 20$ and a scale-free degree distribution. The blue curve corresponds to the theoretical critical point in RBN.

shown), validating a higher correlation of nodes even at a high order, which could be an important evolutionary principle related to the abundance of real networks with these characteristics.

The concepts of synergy and redundancy do not have a unique definition in the literature, so it is important to note that we do not attempt to encompass their whole potential interpretation. Rather, we work with them following their conceptualization through *O-information*. In particular, in Boolean networks, canalization has been identified as a fundamental principle of redundancy, in addition to playing an important role in the criticality of dynamics [5, 15]. Future work will attempt to address canalization specifically, as an additional lever involved in the generation of redundancy.

Having defined the RBN phase diagram (and possibly the phase diagram of any dynamic system) in terms of high-order interdependencies, the next section attempts to specify the fundamental structural characteristics for the generation of redundancy and synergy in each dynamic regime.

4 Phase Diagram Anatomy

To begin to understand how information is effectively processed in each dynamic regime, a possible route is the study of particular network structures. It should be noted that, since this section moves away from the statistical approach of RBNs, focusing on the dynamic effects of certain specific components, here we will consider \mathbf{X}^n as a description of stationary behavior of the system, that is, restrictions to orbits in attractors of the system will be taken. To ensure the uniform contribution of each orbit to \mathbf{X}^n , the duration of each one will correspond to the *least common multiple* of the period of the attractors in the network.

4.1 Ordered Regime

To start exploring the anatomy of the ordered regime, we consider structures where the nodes have exactly one input ($K = 1$). In this case the dynamics is ordered regardless the value of p , and as we see in Fig. 1a, it is always redundant. This restriction induces a particular topology that allows a single circuit of length c from which branches of additional nodes can emerge. Each of these branches will be called chains. Figure 3 presents examples.



Fig. 3. Boolean networks with $n = 10$ and $K = 1$. The colors on the nodes correspond to different redundancy families.

This structure is known as Feed Forward Loop (FFL) and, despite its structural simplicity, it has been widely studied in the literature [4, 12, 17] for its significant concentration and relevance in the dynamics of real biological networks. Mainly, FFLs are considered as necessary design components to improve the robustness of biological systems such as human signaling networks, genetic regulation networks, and others, favoring adaptation against possible external variations. Thus, FFLs structures can strengthen external signal processing or play the role of molecular clocks, a necessary condition for the existence of biological rhythms such as circadian rhythms. This quality of synchronization is strongly validated in [17], by means of a genetic algorithm that finds in the FFLs the core of rhythm generation (with the size of the ring matching the period of the attractors), operating as coordinator of the propagation of state changes.

Here, we will show how the FFL gives rise to redundancy, and thus to the robustness attributed to it in the literature. To do this, first consider an FFL without chains, that is $c = n$. It is not difficult to see that in such a structure there are no transients, therefore \mathbf{X}^n is uniform with respect to each of the 2^c possible states. With these conditions the calculation of the O-information is trivial, since the total entropy, the individual entropies, and the residuals information are maximum: $\sum_n H(X_i) = H(\mathbf{X}^n) = \sum_n R(X_i) = c = n$, and with this $C(\mathbf{X}^n) = B(\mathbf{X}^n) = \Omega(\mathbf{X}^n) = 0$, and there is no high-order interdependency between the nodes of the ring.

We now add chains in the analysis. It is useful to understand a circuit as a signal propagator through the chains (analogously to [17]). Consider $X_{i \rightarrow}$, with

$i \in [1, \dots, c]$, as the set of all nodes that receive the same signal from the sender node i . We will say that the set $X_{i \rightarrow}$ corresponds to a redundancy family, since the nodes that compose it maintain an identical dynamic (or complementary in case of receiving the inverted signal as a consequence of negative edges) and accessing one of them is enough to retrieve the total information at any time t , that is, they share redundant information. To visualize these redundancy families in Fig. 3 they are labeled with different colors. There are c redundancy families, which start on the ring, and have the actual redundancy in the chains that receive the signal. As expected, this dynamics is reflected in $C(\mathbf{X}^n)$. Regardless of the Boolean function applied by each node, if $K = 1$ the network is a conservative system, that is, the number of 1's and 0's in each node and at all times t is constant and in fact balanced. With this $H(X_i) = 1, \forall i \in [1, \dots, n]$, therefore the sum of individual entropies is $\sum_n H(X_i) = n$. On the other hand, since the chains do not interfere with the rhythm of the signal, but rather preserve it until dying out or finding a new subsystem to feed (for cases $K > 1$), the total entropy $H(\mathbf{X}^n)$ remains constant at c even when chains are included. Thus, $C(\mathbf{X}^n) = n - c$, matching the amount of chain nodes. This phenomenon sheds light on the evolution of networks in [17]. There, in the search for robustness, often the genetic algorithm favored the early creation of long chains, prior to the generation of a dominant FFL, since it is in such chains that redundancy lies.

Ensuring the predominance of redundancy in the structure means that we should also be able to calculate the effects of chains on $B(\mathbf{X}^n)$. For this we require the calculation of the sum of residual information $\sum_n R(X_i)$. A chain node is necessarily part of some redundancy family of a ring node, therefore all information contained in it can always be recovered by looking at any other node of the same color. Thus, the residual information of the string nodes is null: $R(X_i) = 0$ for $i \in [c + 1, \dots, n]$. Obviously, the information of the nodes of the ring that form a redundancy family with at least one additional node of the chains, can also be completely recovered without the need to access them directly. On the contrary, if a ring node is unique in its redundancy family, its dynamic information will also be unique and with it $R(X_i) = 1$. The following lemma gives the value of the O-information:

Lemma 1. *Given a Boolean network where $K = 1$ for all its nodes,*

- $C(\mathbf{X}^n) = n - c$, with c the number of nodes that belong to the circuit.
- $B(\mathbf{X}^n) = c - u$, with u equal to the number of unique information nodes belonging to the circuit. In other words, the synergic component of the network $B(\mathbf{X}^n)$ corresponds to the number of different colors present in the chains.
- Since the number of colors in the chains is at most the total number of chain nodes, $\Omega(\mathbf{X}^n) \geq 0$. Hence, if something dominates, it is redundancy.

With this, redundancy as a design principle of FFLs is ratified, but with a specification: high-order interdependencies are only effective in the chains, despite being induced by the circuit. Although the redundant component of this structure can be infinitely increased with the adherence of new chain nodes, the synergistic component is limited by the size of the circuit $B(\mathbf{X}^n) \leq c$.

Example 1. Let \mathbf{X}_1^n y \mathbf{X}_2^n be the stationary behavior produced by the structures in Fig. 3a and 3b respectively.

- (a) Nodes 6, 7, 8, 9, and 10 increase both synergy and redundancy: $C(\mathbf{X}_1^n) = B(\mathbf{X}_1^n) = 5$, therefore there is no predominance and the *O-information* is balanced: $\Omega(\mathbf{X}_1^n) = 0$.
- (b) Nodes 5, 6, 7, 8, 9 and 10 increase redundancy: $C(\mathbf{X}_2^n) = 6$, while redundancy families $X_{2\rightarrow}$, $X_{3\rightarrow}$ and $X_{4\rightarrow}$ increase synergy: $B(\mathbf{X}_2^n) = 3$. Therefore there is a predominance of the redundant component: $\Omega(\mathbf{X}_2^n) = 3$.

Note that considering $K = 1$ implies that $0 \leq \Omega(\mathbf{X}^n) \leq n - 2$, reaching the maximum theoretical value of $\Omega(\mathbf{X}^n)^2$ when $c = 1$ and $n > c \implies B(\mathbf{X}^n) = 1$, $C(\mathbf{X}^n) = n - 1$. On the contrary, the minimum value of $\Omega(\mathbf{X}^n)$ in this case is achieved with the balance between redundancy and synergy ($B(\mathbf{X}^n) = C(\mathbf{X}^n) \approx 0$) when each chain node represents a different redundancy family. It is interesting that in this characteristic scenario of the ordered regime, the calculation of $\Omega(\mathbf{X}^n)$ does not even depend on the functions that govern the dynamics, an appropriate situation to understand the origin of the interdependencies dominated by the redundancy (in addition to providing an analytical description regarding the dynamic relevance of FFLs structures in biological networks). However, this insensitivity to the type of function should vanish as more complex structures are considered, since years of literature give a fundamental role to Boolean functions and the signs of interactions [19]. This is what the next section and future works should elucidate: what richness of dynamics the *O-information* can capture, as the complexity in the networks increases.

4.2 Critical Point

The analysis of the optimal processing regime is harder than the ordered regime: just as the dynamics is more complex, the study of interdependencies is also more complex. To face this problem, the analysis is reduced to a family of motifs with a particularly high prevalence in this type of networks: Coupled Feed Backward Loops (CFBLs) (see Fig. 4). We believe that by beginning to understand how information is propagated within a CFBL, we can give an idea of the origin of the balance between synergy and dynamic redundancy, thereby understanding a little more about the nature of complexity in Boolean networks.

Two circuits are said to be coupled when they share nodes with each other, and in particular, CFBLs are characterized by having circuits with opposite directions. Furthermore, a coupled structure is considered coherent when both circuits maintain the same sign, and incoherent otherwise, following the principles of signed circuits used in [19]. There are previous studies that validate the importance of coherent coupled circuits in the dynamics of biological networks. For example, [9] offers evidence that, when CFBLs are positive, they improve

² Being in fact the only structure capable of achieving this, since the union set of attractors corresponds precisely to two fixed points with maximum Hamming distance.

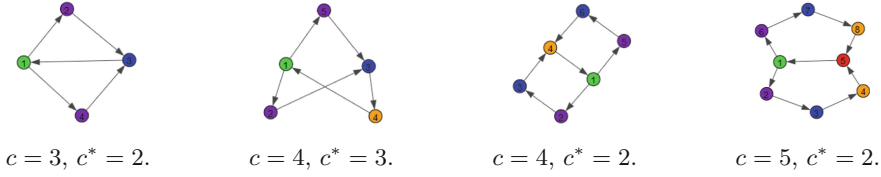


Fig. 4. CFBLs with circuit of size c and c^* nodes shared with each other. Colors label different redundancy families.

signal amplification, while if they are negative they promote homeostasis. Furthermore, coherence in CFBLs has been identified as a design principle in human signaling networks by favoring greater robustness in dynamics [10].

Table 1. Metrics for CFBLs, with circuits of size c and c^* shared nodes.

Structure	Coherence	$\Omega(\mathbf{X}^n)$	$B(\mathbf{X}^n)$	$C(\mathbf{X}^n)$
$c = 3, c^* = 2$	+	0.001	0.981	0.982
	-	0.000	1.000	1.000
$c = 4, c^* = 3$	+	0.001	0.989	0.990
	-	0.000	1.000	1.000
$c = 4, c^* = 2$	+	0.006	1.909	1.915
	-	0.011	2.000	2.011
$c = 5, c^* = 2$	+	0.009	2.801	2.810
	-	0.053	2.991	3.044

We computed the values of $\Omega(\mathbf{X}^n)$, $B(\mathbf{X}^n)$, and $C(\mathbf{X}^n)$ for small examples of CFBLs. A relevant characteristic is that these metrics are exclusively influenced by the sign of the circuits that compose it. In fact, there is a dynamic irrelevance of incoherent CFBLs with respect to high-order interdependencies, since $\Omega(\mathbf{X}^n) = B(\mathbf{X}^n) = C(\mathbf{X}^n) = 0$. Thus, the complex nature of the networks that contain this structure should only depend on their condition of coherence. Table 1 presents the results of calculating the metrics on the four variants of CFBLs in Fig. 4, considering positive and negative coherence in each case. The general trend is effectively a balance between synergy and redundancy with $B(\mathbf{X}^n) \approx C(\mathbf{X}^n) \neq 0$. The value of the synergic component in each case seems to be explained with a similar approach to that used in simple FFLs, since $B(\mathbf{X}^n)$ is limited by the number of redundancy families between nodes not shared by both circuits. Indeed, when coherence is negative, the values of $B(\mathbf{X}^n)$ reach values very close to their structural maximum, while when coherence is positive, the magnitude of the interdependencies decreases slightly. A consequence of increasing the size of the structure by means of non-shared nodes is precisely

the increase of the structural synergistic maximum, and that the magnitude gap between positive and negative coherence is gradually increased.

Table 2. $\Omega(\mathbf{X}^n)$ and its components as a function of the depth of the emerging chains of the CFBL with $c = 4$ and $c^* = 3$.

l	Coherence	$\Omega(\mathbf{X}^n)$	$B(\mathbf{X}^n)$	$C(\mathbf{X}^n)$
1	+	0.001	1.977	1.979
	-	0.000	2.000	2.000
2	+	0.002	2.966	2.967
	-	0.000	3.000	3.000
3	+	0.002	3.954	3.956
	-	0.000	4.000	4.00
4	+	0.991	3.954	4.945
	-	1.000	4.000	5.000

While there is still work to be done to understand how CFBLs communicate with the rest of the network, the analysis described below offers preliminary ideas. Table 2 presents $\Omega(\mathbf{X}^n)$, $B(\mathbf{X}^n)$ and $C(\mathbf{X}^n)$ for the CFBL with $c = 4$ y $c^* = 3$, depending on the depth l of the chains emerging from it. Following the idea of node coloring, the only ones with residual information are those shared by both circuits, since the information of the non-shared ones can be recovered when accessing the simile of the other circuit. With this, the chains emerging from the nodes located immediately before the intersection between both circuits are those that particularly favor the balance between synergy and redundancy (which is where the chains are considered to obtain the values in Table 2). Similar to the case of simple FFLs, the synergy of a Boolean network is limited by the size of the circuits, and its maximum can be reached by a depth of chains equal to the number of shared nodes (if deeper, redundancy increases without any compensation to decrease residual information). Negative coherence precisely makes it possible to achieve the maximum structural synergy, while positive coherence maintains slightly lower values. It should be noted that in scenarios close to the critical point of the phase diagram (where $\bar{K} > 1$), it is unusual to find very long chains, so the synergistic-redundant balance caused by CFBLs is statistically valid and suggestive. In fact, the networks in this regime that we have found often correspond to coupled structures connected to each other by very short chains (if they exist).

We are not claiming coupled structures as a necessary (or sufficient) condition for critical behavior in RBNs; rather, we want to stress their frequent presence (and relevance) in this regime. Finally, and for the sake of completeness: the study of characteristic structures of the chaotic regime is left for future work.

5 Conclusions

The work presented here serves two purposes: to rediscover RBNs' phase diagram in terms of high-order statistical interdependencies (thereby validating *O-information* as a tool for analyzing heterogeneous dynamical systems), and to delve into structural aspects characterizing the ordered regime and the edge of chaos. Specifically, it was determined that the ordered regime is characterized by redundant shared information, while the chaotic regime is characterized by synergistic shared information. Furthermore, the critical point between the two regimes seems to be well defined by a balance between redundancy and synergy. It is also interesting that such behaviors seem to be conserved despite variations in network topologies, as evidenced at least for scale-free networks. On the other hand, the robustness of the interdependencies against perturbations in the initial configurations for each regime is quantified, finding in the *O-information* a fairly consistent dynamic robustness metric. All of the above suggests exploiting this statistical approach offered by RBNs in other widely studied variants such as the diversification of topologies, the effect of network size, restriction to particular Boolean functions, use of asynchronous update schedules, etc.

On the other hand, FFLs are postulated as the fundamental redundancy generation structures in Boolean networks. Likewise, CFBLs seem to correspond to fundamental components in the generation of complex dynamics associated with the critical point between order and chaos, where the quality of *coherence* seems to be relevant for information processing. This also opens up the possibility of using *O-information* as a tool for analyzing the dynamics induced by motifs. In work in progress, we use this approach to shed light on the roles of individual nodes with respect to high-order interdependencies of a Boolean network.

Finally, we want to stress that *O-information* may be used to any dynamic system on complex networks (beyond the Boolean case). However, there is still work to be done to control the computational complexity of its evaluation (or estimation), before it can be used as a practical analysis or design tool on large volumes of data (beyond the theoretical guarantees presented in [20]).

References

1. Aldana, M.: Boolean dynamics of networks with scale-free topology. *Phys. D* **185**, 45–66 (2003)
2. Brillouin, L.: The negentropy principle of information. *J. Appl. Phys.* **24**, 1152–1163 (1953)
3. Derrida, B., Pomeau, Y.: Random networks of automata: a simple annealed approximation. *Europhys. Lett.* **1**, 45–49 (2007)
4. Drossel, B.: Random boolean networks. In: *Reviews of Nonlinear Dynamics and Complexity*, Vol. 1, Ed. H.G. Schuster, Wiley (2008)
5. Gates, A.J., Rocha, L.M.: Control of complex networks requires both structure and dynamics. *Sci. Rep.* **6**(1), 1–11 (2016)
6. Griffith, V., Koch, C.: Quantifying synergistic mutual information. In: *Guided Self-Organization: Inception*. Springer (2014)

7. James, R., Ellison, C., Crutchfield, J.P.: Anatomy of a bit: information in a time series observation. *Chaos* **21**, 037109 (2011)
8. Kauffman, S.A.: Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.* **22**, 437–467 (1969)
9. Kim, J.R., Yoon, Y., Cho, K.H.: Coupled feedback loops form dynamic motifs of cellular networks. *Biophys. J.* **94**, 359–365 (2008)
10. Kwon, Y.K., Cho, K.H.: Coherent coupling of feedback loops: a design principle of cell signaling networks. *Bioinformatics* **24**, 1926–1932 (2008)
11. Langton, C.G.: Computation at the edge of chaos: phase transitions and emergent computation. *Phys. D* **42**, 12–37 (1990)
12. Le, D.H., Kwon, Y.K.: A coherent feedforward loop design principle to sustain robustness of biological networks. *Bioinformatics* **29**, 630–637 (2013)
13. Lizier, J., Prokopenko, M., Zomaya, A.: The information dynamics of phase transitions in random boolean networks. In *Proceedings of 11th International Conference on the Simulation and Synthesis of Living Systems (ALife XI)*. MIT Press (2008)
14. Lloyd-Price, J., Gupta, A., Ribeiro, A.S.: Robustness and information propagation in attractors of random boolean networks. *PLoS One* **7**(7), e42018 (2012)
15. Marques-Pita, M., Rocha, L.: Canalization and control in automata networks: body segmentation in *Drosophila melanogaster*. *PLoS One* **8**(3), e55946 (2013)
16. Oosawa, C., Savageau, M.A.: Effects of alternative connectivity on behavior of randomly constructed Boolean networks. *Phys. D* **170**, 143–161 (2002)
17. Philipp, R., Di Paolo, E.A.: The circular topology of rhythm in asynchronous random Boolean networks. *Biosystems* **73**, 141–152 (2004)
18. Ribeiro, A., Kauffman, S.A., Lloyd-Price, J., Samuelsson, B., Socolar, J.: Mutual information in random Boolean models of regulatory networks. *Phys. Rev. E* **77**, 011901 (2008)
19. Richard, A.: Positive and negative cycles in Boolean networks. *J. Theor. Biol.* **463**, 67–76 (2019)
20. Rosas, F., Mediano, P., Gastpar, M., Jensen, H.: Quantifying high-order interdependencies via multivariate extensions of the mutual information. *Phys. Rev. E* **100**, 032305 (2019)
21. Williams, P.L., Beer, R.D.: Nonnegative decomposition of multivariate information. arXiv preprint [arXiv:1004.2515](https://arxiv.org/abs/1004.2515) (2010)

Earth Sciences Applications



Complexity of the Vegetation-Climat System Through Data Analysis

Andrés F. Almeida-Ñauñay¹(✉), Rosa M. Benito², Miguel Quemada^{1,3},
Juan C. Losada², and Ana M. Tarquis^{1,2,4}

¹ Centro de Estudios e Investigación para la Gestión de Riesgos Agrarios y Medioambientales, CEIGRAM, Universidad Politécnica de Madrid, Senda del Rey, 13, 28040 Madrid, Spain
{af.almeida,miguel.quemada,anamaria.tarquis}@upm.es

² Complex Systems Group, Universidad Politécnica de Madrid,
Avda. Puerta de Hierro, n° 2-4, 28040 Madrid, Spain
{rosamaria.benito,juancarlos.losada}@upm.es

³ Department of Agricultural Production, ETSIAAB, Universidad Politécnica de Madrid,
Avda. Puerta de Hierro, n° 2-4, 28040 Madrid, Spain

⁴ Department of Applied Mathematics, ETSIAAB, Universidad Politécnica de Madrid,
Avda. Puerta de Hierro, n° 2-4, 28040 Madrid, Spain

Abstract. Grasslands in the Iberian Peninsula are valuable and susceptible ecosystems due to their location in arid-semiarid regions. Remote sensing techniques have potential for monitoring them through vegetation indices (VIs). The Modified Soil Adjusted Vegetation Index (MSAVI) is an improved version of classical VIs for arid and semiarid regions.

This work aims to analyse the relation among MSAVI, temperature (TMP) and precipitation (PCP) to understand the complexity of the vegetation-climate system. First, based on MSAVI pattern several phases through the year cycle are defined. Second, a cross-correlation between MSAVI and climatic variables series are performed for each phase at different lags to detect the highest correlation. Then, recurrence plots (RPs) and recurrence quantification analysis (RQA) are computed to characterize and quantify the underlying non-linear dynamics of the MSAVI series.

Our results suggest that five different phases can be defined, in this case study, in which TMP is the main driving factor. The correlation with TMP presents different signs depending on the phase. However, PCP plays a key role with a positive correlation regardless the phase. In the case of TMP, the correlations are higher and the lags shorter than PCP case. This explains the complexity of vegetation-climate dynamics.

RPs and RQA demonstrated to be a suitable tool to quantify this complexity. In our case, we have detected a high-dimensionality and a short-term predictability in the MSAVI series, characteristic of ecological systems.

Keywords: Time-series correlation · Recurrence plots · MSAVI

1 Introduction

Grasslands areas account for 40% of the terrestrial earth surface and cover more than 5 million of hectares in the Iberian Peninsula. In Spain, grasslands represents one of the most beneficial ecosystems for different purposes: biodiversity, meat production, landscape, preservation of traditional values and rural population fixation.

Due to the low cost and real-time data acquisition, remote sensing (RS) techniques have been acknowledged as an appropriate tool to monitor ecosystems. They are based on obtaining reflectance information from the surface properties. Particularly, vegetation has a distinctive response in the near infrared (740–1110, 1300–2500 nm) and the visible (400–700 nm) areas of the electromagnetic spectrum [1].

Vegetation indices (VIs) are mathematical combinations of two or more selected reflectance bands related to biochemical and biophysical vegetation parameters [2]. However, VIs do not identify vegetation activity well in situations when bare soil represents a large part of the surface to be analysed, as it is the case of arid and semi-arid areas. Modified Soil Adjusted Vegetation Index (MSAVI) [3] was developed as a solution for these cases; including a soil adjustment factor.

VIs series present time cycles allowing to describe agro-environmental systems dynamics. Previous studies indicate that vegetation indices behaviour is controlled by climatic fluctuations and have revealed a delayed response of vegetation growth, as a result of the interactions between climate and vegetation [4]. In this line, researchers suggest that there are different lags depending on the variable, ranging from one to two months [5]. In our case, the cross-correlation method allowed to measure the correlation between vegetation indices and climate variables at different lags. Through this analysis, an optimal lag (τ) is obtained; where the correlation between climate variable and vegetation index is maximum.

Vegetation-climate systems present nonlinear characteristics as in any complex system. In 1987, [6] introduced recurrence plots (RPs), which are a simple way to visualize the periodic or chaotic behaviour of a dynamical system through its phase space. There are several works in the framework of RPs and VIs. As an example, [7] applied RPs to measure the determinism and predictability of the NDVI series and its spatial patterns.

This work aims to understand the complex dynamic of pasture-climate system in a semi-arid area. Then, MSAVI temporal dynamics, of this area, are identified through recurrence plot (RP) and the recurrence quantification analysis (RQA).

2 Material and Methodology

2.1 Study Case and Plot Selection

Soto Del Real, Madrid (Spain), named as ZMA, was the pasture area selected for this work. This site is a characteristic Mediterranean climate with warm summers, scarce precipitation, and cold winters. The study area is located on the hillsides of Guadarrama Sierra (Central Spain), where soil materials such as granites and gneiss are predominant. ZMA is situated at 958 m.a.s.l. and the average slope is of 4,7%. ZMA soil is a *Dystric Cambisol*, with a topsoil (0–15 cm) with sandy loam texture, 3% organic matter and a

pH of 5.6. Average precipitation and average temperature are of 541 mm and 13.6 °C, respectively.

The dominant vegetation in the area is Mediterranean grasslands that grow during spring and autumn. They have a summer senescence period and vegetative winter dormancy. Pasture vegetation is grazed by cows and sheep during the whole year with different intensities depending on pasture production, exposing a mix of bare soil and vegetation (live or dead).

Pasture plots were selected based on three criteria: i) maximum area covered by pasture grassland with no woodland, ii) continuous pastureland practices during the analysed period and iii) pastureland cover in the contiguous area. Finally, three plots of 500 × 500 m between (4°32'00" W, 4°33'00" W) and (40°37'00", 40°39'00" N) were selected.

2.2 Acquisition of Satellite Data and MSAVI Calculation

To analyse the pasture cover dynamics, through reflectance measurements, Terra (EOS AM-1) satellite was chosen. MOD09A1 product was selected for this study. This product is a level-3 composite of 500-m resolution imagery. The best pixel observation is chosen within an 8-day period [8].

Study plot reflectance was monitored from 2002 to 2018. Each year, 46 images were acquired, giving a total amount of 782 images in the study period. Two spectral bands are extracted from the imagery collection: RED (620–670 nm) and NIR (841–876 nm). To assure a correct spectral characterisation of the area, an average of each band from the three plots was calculated.

Spectral index sensitivity is greatly affected by soil brightness. For this purpose, MSAVI includes a soil factor adjustment (L_M) dependant on the local conditions [3]. L_M is calculated by the following formula:

$$L_M = \frac{2 * s(NIR - RED) * (NIR - s * RED)}{(NIR + RED)} \quad (1)$$

where s is defined as the soil line given by a plot of RED vs. NIR brightness. As reported by [9], the soil line (s) is expected to be the baseline to estimate vegetation indices that include the soil background in their calculation. In this work, [10] and [11] methods were combined to obtain the soil line.

Once L_M is calculated using equation [1], to estimate MSAVI the following equation is applied:

$$MSAVI = \frac{NIR - RED}{NIR + RED + L_M} * (1 + L_M) \quad (2)$$

2.3 Meteorological Variables

An AEMET (*Agencia Estatal de Meteorología*) station was used to obtain daily meteorological data. The meteorological station is sited between Soto del Real and Colmenar Viejo (40° 41' 46.008" N, 3° 45' 54.019" W) at 1004 m.a.s.l. From this station average

daily air temperature (T_m) and daily precipitation were obtained. These variables were transformed in a series in which TMP was the average of T_m each 8 days and PCP was the accumulated daily precipitation during 8 days. The length of these series was the same than MSAVI series.

2.4 Date-to-Date Analysis

To characterize the MSAVI behavior a descriptive statistic had been applied per date using box-plots charts. Several phases can be discriminated based on the visual observation of trend changes in the MSAVI series; that compose an annual cycle (Table 1).

Table 1. Annual pasture phases based on MSAVI trend at ZMA (Madrid).

Initial date	Final date	Code phase	MSAVI trend
Nov. 25th	Jan. 25	P0	Constant
Feb. 2	Apr. 23	P1	Increasing
May. 1	Jun. 20	P2	Decreasing
Jul. 28	Sep. 22	P3	Constant
Sep. 30	Nov. 17	P4	Increasing

Based on these phases, linear regressions and Pearson's coefficients analysis were conducted to show the relationship between vegetation indices and climate variables.

2.5 Cross-Correlations by Phase

VI dynamics fluctuate depending on the season of the year. This indicates that a constant optimum time lag (τ) through all the year might be inadequate. With taking into account the differing time lags for each climate factor, the relationship between the MSAVI and climate variables was analysed through correlation coefficients, calculated by the following equation:

$$P_{x,y} = \frac{\sum_{i=1}^N \{(x_i - \bar{x})(y_i - \bar{y})\}}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2 \sum_{i=1}^N (y_i - \bar{y})^2}} \quad (3)$$

Being N the number of the years, x the MSAVI series and y the climate variable series. Then, correlation between MSAVI and climate time-series was calculated in each one of the above-mentioned phases, using an accumulative 8-days lag period during the phase duration.

2.6 Recurrence Plots and Recurrence Quantification Analysis

Recurrence plots (RP) allow to visualize system states in the phase space. In complex dynamical systems, recurrence is a concept related to the temporal evolution of dynamical systems trajectories in the phase space.

Generally, to create a RP an embedding dimension (m) and a time-delay (τ) are necessary. Delay, τ , is the minimum time lag to minimize the autocorrelation of a time series. Then, m represents the number of independent variables needed to characterize the system. Mathematically RP is defined as:

$$R_{ij} = \Theta(\varepsilon - \|\vec{x}_i - \vec{x}_j\|), \vec{x}_i \in \mathbb{R}^m, i, j = 1 \dots N, \quad (4)$$

where N is the number of measured states \vec{x}_i , Θ is the Heaviside step function (i.e. $\Theta(x) = 1$, if $\|\vec{x}_i - \vec{x}_j\| \leq \varepsilon$, and $\Theta(x) = 0$ otherwise), $\|\cdot\|$ is a norm and ε is a threshold previously defined based on the time-series properties. In this study, the phase space trajectories are based on the Euclidean distance between \vec{x}_i and \vec{x}_j of the series. If $R_{ij} = 1$ at a time (i, j), is marked as a black dot in the position (i, j). Otherwise, if $R_{ij} = 0$ recurrence states will be represented as white dots.

Recurrence Quantification Analysis (RQA) is based on the quantification of the small-scale structures in RPs [12]. Several measures of complexity have been proposed, however, in this work we focused on: Determinism (DET), Average length of structures (LT), Shanon Entropy (ENT) and Laminarity (LAM).

The CRQA R package [13], based on the Cross Recurrence Plot Toolbox developed by [14], was used to construct RP and obtain RQA measures. First, MSAVI series were normalized using z-score and distance matrix was rescaled based on the maximum value following the recommendations of [15] and [16]. *Optimizeparam* function was then computed to found the optimal values of the three parameters (τ , m , and ε). The delay (τ) is obtained by finding the local minimum where mutual information drops to both series. The embedding dimension (m) is determined by the false nearest neighbours' algorithm. The threshold ε is estimated by an iterative process based on the standard deviation (SD) of the time series.

The quantification of RP structures was computed with the *Crqa* function using the three values obtained from the optimization.

3 Results and Discussion

3.1 Box Plots and Phases Analysis

Box plots dispersion of MSAVI and TMP are displayed in Fig. 1A. The MSAVI highest dispersion is located in P2 reaching stable and less dispersed values during P3. A similar trend is reported by [17] in the case of Normalized Difference Vegetation Index (NDVI) from MODIS. However, we found much lower values for MSAVI in the dry season, from the beginning of June until the end of September due to the arid-semiarid climate in this study. As we can see in Fig. 1B, the rain is almost inexistent.

Temperature dispersion is higher in P1 and P2, being more stable in P4 and P0. The highest precipitation dispersion (Fig. 1B) is located at the same phases in which MSAVI is more disperse (P1 and P2).

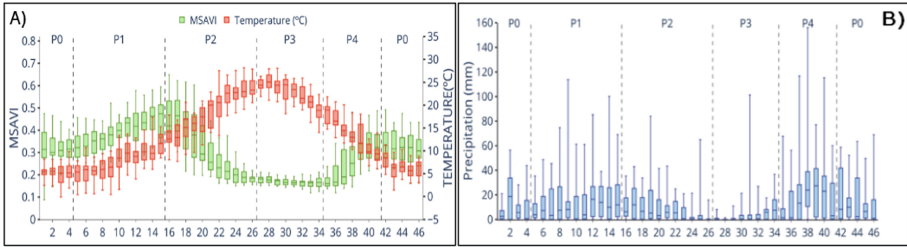


Fig. 1. Box plots of MSAVI and average temperature (A) and accumulated precipitation (B), 8-day period at ZMA (Madrid).

Based on box plots results, linear regression analysis is conducted to study the relation of each climatic variable in each phase with MSAVI values (Fig. 2).

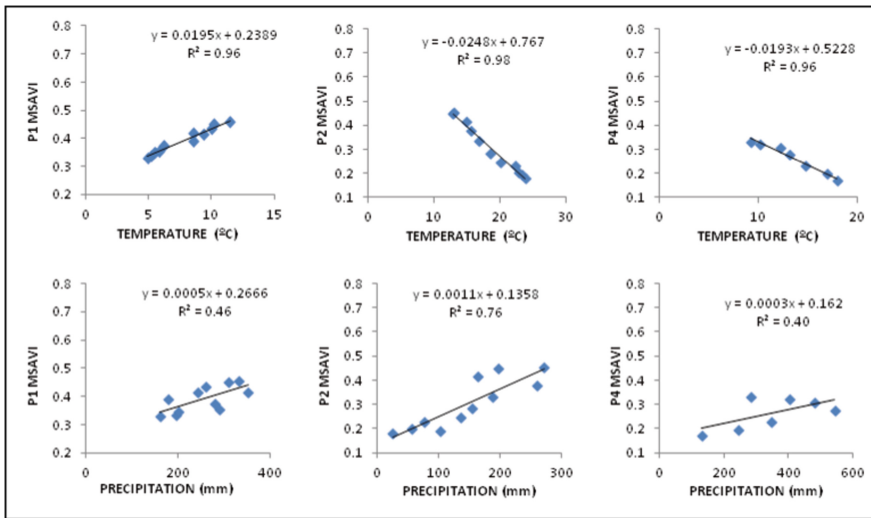


Fig. 2. Relation of MSAVI to average temperature (TMP) and accumulated precipitation (PCP) at 8-day period during phases 1 (first column), 2 (second column) and 4 (third column).

In general, the temperature is identified as the main driving factor in the vegetation-climate system; as it shows high R² values (>0.9) in all the phases. Precipitation shows lower R² values in comparison to temperature; being the highest (>0.7) in P2. It is important to note that temperature trend varies depending on the phases, being positive on P1 and negative in P2 and P4 phases. In the meantime, precipitation has the same trend; being positive in all the phases and pointing out that precipitation is regularly favourable in semiarid-grassland growth. This fact is in line with previous work, such as [18] that revealed a positive relationship between the amount of precipitation and the net primary grasslands production.

3.2 Cross-Correlation by Phase

A fluctuating lag is observed in cross-correlation coefficients pointing out the complexity of the pasture-climate dynamics along a year (Table 2).

The lags when the correlation is maximum with MSAVI varied between 0 to -3 depending on the phases and variables. The most correlated variable is temperature. This variable shows a null lag at P1 and P2 increasing to -2 during P4. On the other hand, precipitation shows lower correlation values and longer lags. In P1 and P4 the lag is -3 for PCP, phases in which temperature values are less than $17\text{ }^{\circ}\text{C}$. During P2, PCP lag is shorter (-1) presenting during half of this phase temperatures closer to $25\text{ }^{\circ}\text{C}$.

Table 2. Cross-correlation coefficients between MSAVI and temperature (TMP) and precipitation (PCP) at different lags in each phase at ZMA (Madrid). Bold letter represents the maximum correlation in each row. Each time lags is of an 8-days period.

		Time lags						
		0	-1	-2	-3	-4	-5	-6
P1	TMP	0.422	0.308	0.259	0.233	0.169	0.109	0.029
	PCP	0.032	0.088	0.186	0.221	0.156	0.102	0.093
P2	TMP	-0.763	-0.757	-0.717	-0.699	-0.695	-0.697	-0.637
	PCP	0.359	0.396	0.381	0.321	0.357	0.309	0.230
P4	TMP	-0.594	-0.600	-0.625	-0.602	-0.603	-0.581	-0.512
	PCP	0.159	0.297	0.393	0.397	0.293	0.275	0.304

3.3 Differencing Vegetation Index Series and Parameter Optimization

A preliminary analysis of the RP, with an embedding dimension $m = 1$ and lag $\tau = 0$, for MSAVI time-series was computed. This graphic is shown in Fig. 3A and represents the normalized VI series against itself. It is observed an isolated point structure which may indicate noisy behavior in the vegetation index time-series. The results of the prior analysis indicate that it is necessary to optimize RP critical parameters to search for a clearer pattern. The *Optimizeparam* function was computed to estimate the parameters of RPs, $m = 7$, $\tau = 9$, $\varepsilon = 20.5977$ and $\text{RR} = 4.53\%$ were selected as the most optimal parameters obtaining a RP showed in Fig. 3B.

At large scale, white stripes, are related to atypical values and an interruption in the vegetation pattern [19]. We believe that this behaviour is due to an extreme climatic event that increased soil moisture; consequently, VI series values atypically increased, as observed in Fig. 3B. At the same time, small-scale structures, periodic patterns (diagonal line shapes) are observed in MSAVI optimized RP, which might represent seasonal variability. This visual inspection is in line with the work of [20] that revealed similar patterns in a northeast grassland zone in Spain.

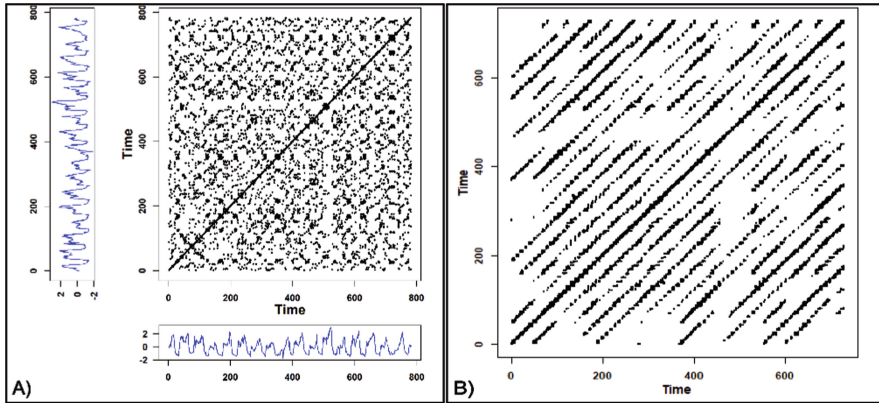


Fig. 3. A) Recurrence plots of MSAVI for ZMA zone. Vegetation indices series is normalized by the z-score method. Time units are represented as the X and Y axis. Each time-unit is a period of 8-days, coincident with 8-days compose MODIS images during the study period (2002–2018). Embedding dimension $m = 1$, delay $\tau = 0$ and recurrence rate $RR = 1.5\%$. B) Optimized recurrence plots using normalized VI data and rescaled distance matrix for ZMA

A higher embedding dimension than expected was obtained. This fact is supported by findings in the literature [21], which relate higher dimensionality to system complexity. Ecological systems present nonlinear dynamics, combining periodic and chaotic cycles, whose equations ruling the systems are unknown. In this line, [20] demonstrated the usefulness of RPs to describe nonlinear behaviours in high-dimensional systems, such as MSAVI time series.

3.4 Recurrence Quantification Analysis

The quantification of RP structures was computed with the Crqa function, and the results are exposed in Table 3.

Table 3. Recurrence Quantification Analysis (RQA) of ZMA = *Soto Del Real*, using rescaled data. RQA of artificial series, adapted from [22], were added for comparison. MSAVI = Modified Soil-Adjusted Vegetation Index, DET = Determinism, LT = Average length of diagonal structures, ENTR = Shannon Entropy, LAM = Laminarity.

Case	DET (%)	LT	ENTR	LAM (%)
MSAVI				
ZMA	75.71	3.89	1.65	85.25
Artificial series				
Stochastic	7.90	2.05	0.20	9.40
Periodic	95.90	11.16	2.20	82.30

Based on the density of recurrence points, determinism has been related to the chaotic or periodic behaviour of the system, representing a measure of temporal stochasticity. Determinism (DET) has been utilized as an indicator of climate stability [7] or the detection of bioclimatic transitions [22]. Our results suggest that a high value of DET is related to an adequate characterisation of pasture vegetation pattern through MSAVI index,

Increases in LT are interpreted as a larger time of predictability, as it has been reported by [23] work. MSAVI LT values obtained are low compared to periodic series, Table 3 indicates that vegetation may be predicted in the short term due to the great complexity of ecological systems reported by [24].

ENTR refers to the disorder of the system. Standard values obtained by [22] noted that stochastic systems tend to obtain lower ENTR values (0.2) in comparison with those of periodic systems (2.20). We speculate that the high value of MSAVI ENTR is the consequence of the high number of precipitations events in the zone. Box plot precipitation shed a light about the water status in the area, in this case, ZMA precipitation is higher than the average of Mediterranean climate. This fact is sustained by [20] findings which suggest that grassland areas with higher precipitations tend to obtain higher ENTR values.

LAM refers to the chaos-chaos transitions and is directly related to the detection of laminar states [25]. MSAVI series presents a high number of laminar states indicating indicates that VI values are trapped during certain time frames, decreasing time series variability and supporting the idea of higher predictability and determinism of MSAVI index.

4 Conclusions

In summary, we have applied the cross-correlation method as a prior step to characterize the complexity of the vegetation-climate system, concluding that temperature is a strong driver factor. However, it is important to note that precipitation showed a stable positive trend along the phases suggesting that precipitation events are beneficial in arid-semiarid grassland, regardless of the time of a year. In addition, it was revealed that lag between MSAVI and climatic series is variable depending on the phase and climatic variable.

Then, RP and RQA were applied to MSAVI time-series to measure the complexity of the pasture-climate system. We detected a characteristic dynamic that point out short-term predictability and high-dimensionality of the MSAVI time series. In the end, this work emphasizes the potential of recurrence plots and recurrence quantification analysis to characterise and quantify the complexity of a vegetation-climate system.

Acknowledgments. The authors acknowledge support from Project No. PGC2018-093854-B-I00 of the Spanish Ministerio de Ciencia Innovación y Universidades of Spain and the funding from the Comunidad de Madrid (Spain), Structural Funds 2014-2020 512 (ERDF and ESF), through project AGRISOST-CM S2018/BAA-4330 and the financial support from Boosting Agricultural Insurance based on Earth Observation data - BEACON project under agreement N° 821964, funded under H2020_EU, DT-SPACE-01-EO-2018-2020.

References

1. Rouse, J.W., Haas, R.H., Schell, J.A., Deering, D.W.: Monitoring the vernal advancement and retrogradation (green wave effect) of natural vegetation. *Prog. Rep. RSC* **1978-1**, 2–8 (1973)
2. Huete, A., Didan, K., Miura, T., Rodriguez, E.P., Gao, X., Ferreira, L.G.: Overview of the radiometric and biophysical performance of the MODIS vegetation indices. *Remote Sens. Environ.* **83**(1–2), 195–213 (2002)
3. Qi, J., Chehbouni, A., Huete, A.R., Kerr, Y.H., Sorooshian, S.: A modified soil adjusted vegetation index. *Remote Sens. Environ.* **48**(2), 119–126 (1994)
4. Guo, B., Zhou, Y., Wang, S., Tao, H.: The relationship between normalized difference vegetation index (NDVI) and climate factors in the semiarid region: a case study in Yalu Tsangpo River basin of Qinghai-Tibet Plateau. *J. Mt. Sci.* **11**(4), 926–940 (2014). <https://doi.org/10.1007/s11629-013-2902-3>
5. Shen, B., Fang, S., Li, G.: vegetation coverage changes and their response to meteorological variables from 2000 to 2009 in Naqu, Tibet, China. *Can. J. Remote. Sens.* **40**(1), 67–74 (2014)
6. Eckmann, J.P., Oliffson Kamphorst, O., Ruelle, D.: Recurrence plots of dynamical systems. *Epl* **4**(9), 973–977 (1987)
7. Li, S.C., Zhao, Z.Q., Liu, F.Y.: Identifying spatial pattern of NDVI series dynamics using recurrence quantification analysis. *Eur. Phys. J. Spec. Top.* **164**(1), 127–139 (2008)
8. LP DAAC: Land processes distributed active archive center: surface reflectance 8-day L3 global 500 m, NASA and USGS (2014)
9. Baret, F., Guyot, G.: Potentials and limits of vegetation indices for LAI and APAR assessment. *Remote Sens. Environ.* **35**(2–3), 161–173 (1991)
10. Xu, D., Guo, X.: A study of soil line simulation from landsat images in mixed grassland. *Remote Sens.* **5**(9), 4533–4550 (2013)
11. Xu, M., Eckstein, Y.: Use of weighted least squares method in evaluation of the relationship between dispersivity and field scale. *Groundwater* **33**(6), 905–908 (1995)
12. Webber, C.L., Zbilut, J.P.: Dynamical assessment of physiological systems and states using recurrence plot strategies. *J. Appl. Physiol.* **76**(2), 965–973 (1994)
13. Coco, M.I., Dale, R.: Cross-recurrence quantification analysis of categorical and continuous time series: an R package. *Front. Psychol.* **5**(1), 1–14 (2014)
14. Marwan, N.: CRP Toolbox 5.22 (R32.4) (2007). <http://tocsy.pik-potsdam.de/CRPtoolbox/>. Accessed 28 June 2019
15. Patro, S.G.K., Sahu, K.K.: Normalization: a preprocessing stage. *Iarjset*, pp. 20–22 (2015)
16. Webber, C.L., Zbilut, J.: Recurrence quantification analysis of nonlinear dynamical systems. In: *Tutorials in contemporary nonlinear methods for the Behavioral Sciences Web Book*, no. June, pp. 26–94 (2005). <http://www.nsf.gov/sbe/bcs/pac/nmbs/nmbs.jsp>. Accessed 5 June 2019
17. Wang, X., Ge, L., Li, X.: Pasture monitoring using SAR with COSMO-skymed, ENVISAT ASAR, and ALOS PALSAR in Otway, Australia. *Remote Sens.* **5**(7), 3611–3636 (2013)
18. Heisler-White, J.L., Knapp, A.K., Kelly, E.F.: Increasing precipitation event size increases aboveground net primary productivity in a semi-arid grassland. *Oecologia* **158**(1), 129–140 (2008)
19. Proulx, R., Parrott, L., Fahrig, L., Currie, D.J.: Long time-scale recurrences in ecology: detecting relationships between climate dynamics and biodiversity along a latitudinal gradient. In: Webber, C.L., Marwan, N. (eds.) *Recurrence Quantification Analysis – Theory and Best Practices*, no. February, pp. 335–347. Springer, Cham (2015)
20. Marwan, N., Kurths, J., Foerster, S.: Analysing spatially extended high-dimensional dynamics by recurrence plots. *Phys. Lett. Sect. A Gen. At. Solid State Phys.* **379**(10–11), 894–900 (2015)

21. Beldare-Franch, J., Contreras, D., Tordera-Lledó, L.: Assessing nonlinear structures in real exchange rates using recurrence plot strategies. *Phys. D Nonlinear Phenom.* **171**(4), 249–264 (2002)
22. Zhao, Z., Liu, J., Peng, J., Li, S., Wang, Y.: Nonlinear features and complexity patterns of vegetation dynamics in the transition zone of North China. *Ecol. Indic.* **49**, 237–246 (2015)
23. Frilot II, C., Kim, P., Carrubba, S., McCarty, D., Chesson Jr., A.L., Marino, A.: Analysis of brain recurrence. In: Webber, C.L., Marwan, N. (eds.) *Recurrence Quantification Analysis – Theory and Best Practices*, no. February, pp. 213–251. Springer, Cham (2015)
24. Beckage, B., Gross, L.J., Kauffman, S.: The limits to prediction in ecological systems. *Ecosphere* **2**(11), 1–12 (2011)
25. Marwan, N., Wessel, N., Meyerfeldt, U., Schirdewan, A., Kurths, J.: Recurrence-plot-based measures of complexity and their application to heart-rate-variability data. *Phys. Rev. E - Stat. Physics, Plasmas, Fluids, Relat. Interdiscip. Top.* **66**(2), 1–8 (2002)



Towards Understanding Complex Interactions of Normalized Difference Vegetation Index Measurements Network and Precipitation Gauges of Cereal Growth System

David Rivas-Tabares¹(✉)  and Ana M. Tarquis^{1,2,3} 

¹ CEIGRAM, ETSIAAB, Universidad Politécnica de Madrid, Madrid, Spain
davidandres.rivas@upm.es

² Complex Systems Group (GSC), Universidad Politécnica de Madrid (UPM), Madrid, Spain

³ Department of Applied Mathematics, ETSIAAB, Universidad Politécnica de Madrid (UPM), Madrid, Spain

Abstract. Earth observations (EO) are nowadays a powerful tool to evaluate vegetation systems as crops to reach Sustainable Development Goals (SDGs) of the agenda 2030. Normalized Difference Vegetation Index (NDVI) is a popular and widespread index in remote sensing to evaluate vegetation dynamics. However, analytical advances of NDVI long term series analysis are towards understanding complex relations of atmosphere-plant-soil system through temporal and scaling behaviour. Hence, this research presents the generalized structure function (GSF) and Hurst exponent as innovative analytical methods to explore a satellite-based network of NDVI measurements and precipitation series in cereals in the semi-arid. Results suggest that weather support anti-persistence structure of NDVI time series since weather regime in semi-arid is essential in the understanding of complex processes of the crop growth. Mathematical description of NDVI series coupled with GSF and Hurst exponent can reinforce crop modelling future purposes.

Keywords: Cereals phenology scaling · Hurst exponent · NDVI time series

1 Introduction

Earth observation time series analysis is increasingly improved for multiple vegetated and unvegetated areas evaluation. However, characterize agricultural land processes coupling to weather are challenging due to multitude of processes and factors affecting vegetation growth. One of these growing factors in semiarid is especially the rainfall behaviour on agricultural fields, in which plant, soil, and climate are strongly correlated with crop yield. These relationships are commonly analysed using vegetation indices such as the normalized difference vegetation index (NDVI).

The analysis of intensive long-term cereal sequences is very scarce from earth observations. Even the NDVI long-term series from monoculture and rotational cereals sequences have not been deeply studied in semiarid, although these remain one driving

factor of soil degradation in those areas [1–3]. Site selection for NDVI spatial analysis can be treated as a network of measurements to capture vegetation phenology variations. Some studies that analysed the relation of scaling behaviour through the mass distribution of land management in crops and soil types (e.g., tillage, land levelling, etc.) introduced a promising method [4] to complement spatial features and environmental insights to mitigate soil degradation. This method for scaling properties is the generalized structure function (GSF). The scaling properties of reflectance signals from satellites can provide complementary information to specific sites [5–7] increasing the temporal understanding of cereal phenology sequences. These scaling properties of reflectance signals, along time series, can be described as a mass distribution on a temporal domain complementing classical statistics of the measured signals [8]. Hence, stationary series from a satellite-based network of cereal NDVI measurements can be related to site weather interactions, especially with precipitation (pcp) patterns.

2 Methods

2.1 Case Study and Data

The study area is located in north-central Spain in the midlands of the Duero River basin, Fig. 1. This area overlaps with most of the Avila and Segovia provinces. The area was delineated using the midlands of the Eresma and Adaja Rivers [9], and it covers 200,197 ha. The land use of the area is mainly rainfed cereal agriculture (70%), of which 41% is barley, 15% is wheat, and 14% are other crops (e.g., canola, sunflower, and peas), as the most typical rainfed crops in the area. Both crops are part of the most representative features of the crop rotation sequence in the area, being the focus of interest in this study.

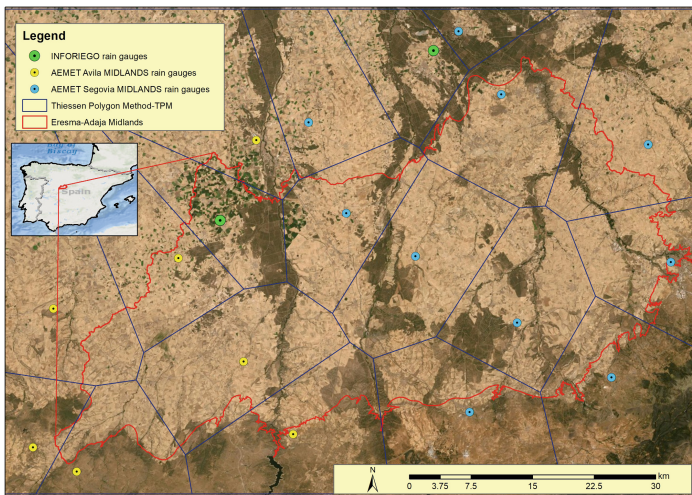


Fig. 1. Study area located in north-central Spain in the midlands of the Duero River basin.

Site selection of cereal plot was based on a previous study in which long-term cereal schemas were identified from remote sensing [9]. These plots comprise several

sub-basins and different soil types. From those, two areas were finally selected as representative for each river in terms of weather and soil type. The selected final areas are shown with color wheat spikes in Fig. 2.

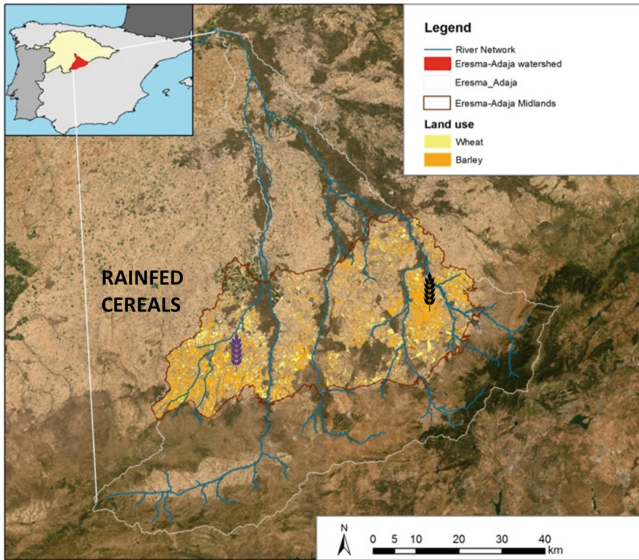


Fig. 2. Location of the plots with long-term cereal sequences NDVI measurements network, black spike (right) indicates the selected site from Adaja River basin and purple spike (left) indicates site from Eresma River basin.

The MODIS-Terra MOD13Q1 V06 product at 250 m spatial resolution and 16-day composite images [10] from 2000 to 2019 (451 images) were used to define NDVI measurement network. The extracted series from MOD13Q1 comprise data from 02-02-2000 to 14-09-2019 that were checked through the quality and reliability pixel index of MODIS data, and only high-quality pixels (rank key = 0) were filtered for the series. It is important to highlight that the 16-day composite NDVI series are generated using the two 8-day composite surface reflectance granules (MOD09A1) in the 16-day period considered one of the most spatiotemporal reliable products of MODIS [10]. Some of the depressed values of the series were preprocessed (i.e., less than 7 values in the series) through the Savitzky–Golay filter [11] to smooth the time series, specifically those that were caused primarily by cloud contamination and atmospheric variability [12]. The data extraction from the MODIS product was performed using the Google Earth Engine [13]. Each site represents the average of five cereal plots in which all years were planted with cereals. Three time series were analyzed for each site, i) NDVI average for the 5 plots, ii), NDVI residual series (the former series subtracted annual pattern) and iii), NDVI anomalies [14] calculated as $ZNDVI = NDVI_i - \mu NDVI / SD$.

Meteorological national data from a network of 17 precipitation gauges [15] were used to set up weather assignment to crop plots. This was performed applying the Thiessen

Polygon Method (TPM) from rain gauges to weighted configuration of subbasin precipitation series [9]. As similar to NDVI series, three time series were analyzed for precipitation, i) Pcp average for subbasin plots, ii), NDVI residual series (the former series subtracted annual pattern) and iii), Pcp anomalies.

3 Results

3.1 Analytical Advances of NDVI Cereal Time Series

Due to the seasonal pattern of NDVI signals, their long-term statistics do not change significantly over time. The statistics of the NDVI series were developed over the tillage period from March (vegetation cover >30%) to June until the grain harvest, which is denoted the growing season in cereals. The confidence interval reveals that in the growing season, the sites are statistically different when the surface is covered by vegetation. The ANOVA results confirm that NDVI values for the sample sites during the vegetative period exhibit significant differences with a confidence level of 99% and p-values < 0.01.

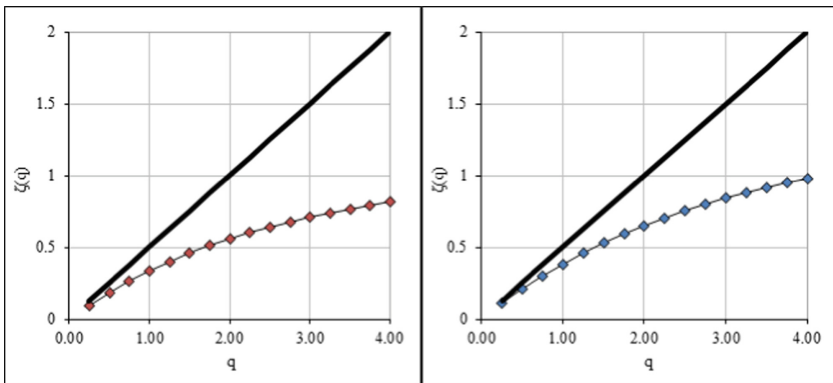


Fig. 3. Generalized Structure Function plots for NDVI residual series of eastern sites (left column) and western sites (right column) of $\zeta(q)$ curve.

The scaling of the NDVI series (original, residual and anomalies) was confirmed through the GSF calculation. The GSFs relate the $S_q(\Delta i/L)$ against the $(\Delta i/L)$ with $L = \Delta i_{max}$ for the NDVI residual series of the sites. Thus, the maximum increment was chosen in 32 data points ($L = \Delta i_{max} = 64$), which is equivalent to a 32-month period or 2 growing seasons. In this case, values for q were selected between 0.25 and 4 with 0.25 increments, Fig. 3. The Hurst exponents curve results from the GSF plot across all the q exponents. For this case this relation reveals that the NDVI residual signals of the sites are anti-persistent in time, Fig. 4. Red dots from west sites and blue dots from east sites. However, the anti-persistence degree of these NDVI residual series between sites is shifted but not statistically different.

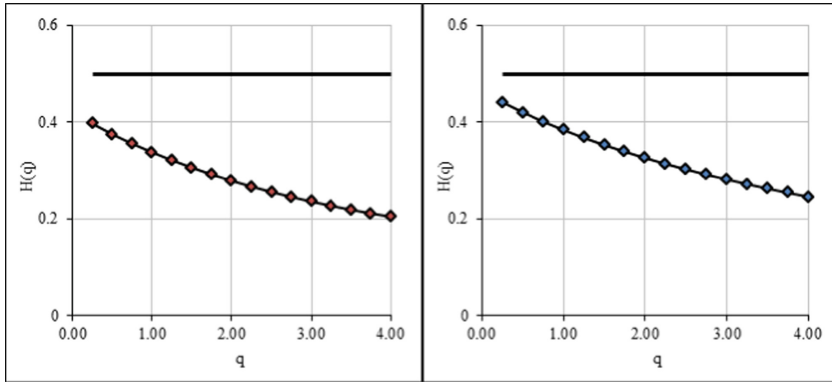


Fig. 4. Generalized Hurst exponent $H(q)$ for NDVI residual series of eastern sites (left column) and western sites (right column). The continuous line correspond to non correlated noise with Hurs value of 0.5.

3.2 Scaling Characteristics of Precipitation Series

The precipitation as the main water source into the system (rainfed condition) was also evaluated using time series every 15 days. For this reason, the precipitation time series for both sites were also analyzed similarly to NDVI series using the GSF, Fig. 5. The curve $\zeta(q)$ and the generalized Hurst exponent $H(q)$ curves also show that the precipitation residual series of the sites present an extreme anti-persistent character.

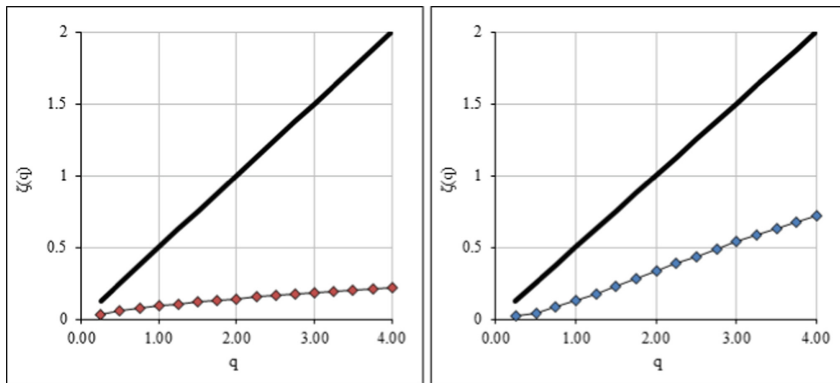


Fig. 5. Generalized Structure Function plots for Pcp residual series of eastern sites (left column) and western sites (right column) of $\zeta(q)$ curve.

The resulting noise exhibited a scaling behavior, and the generalized Hurst exponent was also anti-persistent, Fig. 6. This situation can support the anti-persistent response of NDVI residuals when presenting the anti-persistent noise structure of the precipitation time series. To our knowledge, there is no scientific evidence about the NDVI residual anti-persistent series in conjunction with the precipitation residual anti-persistent series from cereal sequences in semiarid conditions.

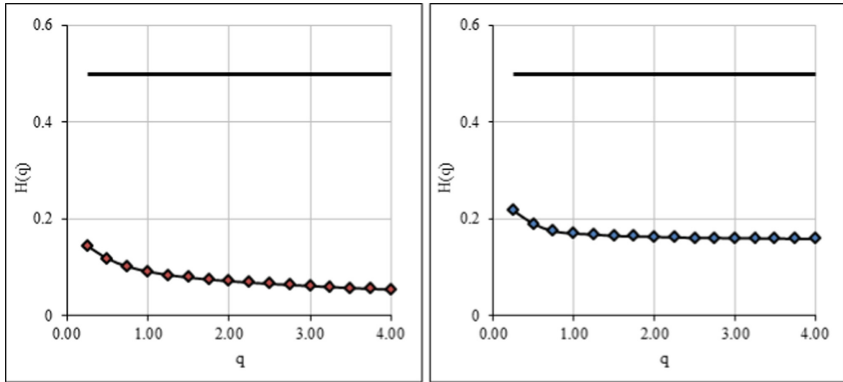


Fig. 6. Generalized Hurst exponent $H(q)$ for Pcp residual series of eastern sites (left column) and western sites (right column). The continuous line correspond to non correlated noise with Hurst value of 0.5.

4 Conclusions

The results presented in this work reinforce the idea that the knowledge of rain gauges spatial variability is a key component in understanding patterns of vegetation at large scales, specifically related to cereal yields in the semiarid. The NDVI residual series under rainfed activity for cereal production in the semiarid climate in Spain exhibits an anti-persistent structure; this is primarily due to the anti-persistent behaviour of precipitation residual series. The time series analysis of vegetation indices, such as satellite-based NDVI measurement network, in combination with precipitation time series from dense rain gauges networks provides some insights into the understanding of seasonal cereal yields. This approach has the objective of obtaining feedback and identifying the field features associated with the anti-persistent structure of NDVI residual time series since weather regime in semi-arid is essential in the understanding of complex processes of the crop growth. Mathematical description of NDVI series coupled with GSF and Hurst exponent can reinforce crop modelling future purposes. Defining NDVI measurement networks constitutes a low-cost and efficient tool to track temporal variations of rainfed cereal dynamics. NDVI time-series provide effective estimates of crop growth states and constitutes accurate estimates of crop timing of main phenological events such as tillering, stem extension, heading and ripening.

Acknowledgements. The authors acknowledge support from Project No. PGC2018-093854-B-I00 of the Spanish *Ministerio de Ciencia Innovación y Universidades* of Spain and the financial support from Boosting Agricultural Insurance based on Earth Observation data - BEACON project under agreement N° 821964, funded under H2020_EU, DT-SPACE-01-EO-2018-2020.

References

1. Mao, R., Zeng, D.-H., Li, L.-J., Hu, Y.-L.: Changes in labile soil organic matter fractions following land use change from monocropping to poplar-based agroforestry systems in a semiarid region of Northeast China. *Environ. Monit. Assess.* **184**, 6845–6853 (2012)

2. Hernanz, J.L., López, R., Navarrete, L., Sanchez-Giron, V.: Long-term effects of tillage systems and rotations on soil structural stability and organic carbon stratification in semiarid central Spain. *Soil Tillage Res.* **66**, 129–141 (2002)
3. Wu, H., Wu, L., Zhu, Q., Wang, J., Qin, X., Xu, J., Kong, L., Chen, J., Lin, S., Khan, M.U.: The role of organic acids on microbial deterioration in the *Radix pseudostellariae* rhizosphere under continuous monoculture regimes. *Sci. Rep.* **7**, 1–13 (2017)
4. Moreno, R.G., Alvarez, M.C., Requejo, A.S., Tarquis, A.M.: Multifractal analysis of soil surface roughness. *Vadose Zo. J.* **7**, 512–520 (2008)
5. Zeleke, T.B., Si, B.C.: Scaling properties of topographic indices and crop yield. *Agron. J.* **96**, 1082–1090 (2004)
6. Wang, Z., Shu, Q., Liu, Z., Si, B.: Scaling analysis of soil water retention parameters and physical properties of a Chinese agricultural soil. *Soil Res.* **47**, 821–827 (2010)
7. Zheng-Ying, W., Qiao-Sheng, S.H.U., Li-Ya, X.I.E., Zuo-Xin, L.I.U., Si, B.C.: Joint multifractal analysis of scaling relationships between soil water-retention parameters and soil texture. *Pedosphere* **21**, 373–379 (2011)
8. Tarquis, A.M., Morato, M.C., Castellanos, M.T., Perdigones, A.: Comparison of structure function and detrended fluctuation analysis of wind time series. *Nuovo Cim. Della Soc. Ital. Di Fis. C Geophys. Sp. Phys.* **31**, 633–651 (2008)
9. Rivas-Tabares, D., Tarquis, A.M., Willaarts, B., De Miguel, Á.: An accurate evaluation of water availability in sub-arid Mediterranean watersheds through SWAT: Cega-Eresma-Adaja. *Agric. Water Manag.* **212**, 211–225 (2019). <https://doi.org/10.1016/j.agwat.2018.09.012>
10. Didan, K.: MOD13Q1 MODIS/Terra vegetation indices 16-day L3 global 250 m SIN grid V006. NASA EOSDIS L. Process. DAAC (2015)
11. Savitzky, A., Golay, M.J.E.: Smoothing and differentiation of data by simplified least squares procedures. *Anal. Chem.* **36**, 1627–1639 (1964)
12. Chen, J., Jönsson, P., Tamura, M., Gu, Z., Matsushita, B., Eklundh, L.: A simple method for reconstructing a high-quality NDVI time-series data set based on the Savitzky-Golay filter. *Remote Sens. Environ.* **91**, 332–344 (2004)
13. Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., Moore, R.: Google earth engine: planetary-scale geospatial analysis for everyone. *Remote Sens. Environ.* **202**, 18–27 (2017)
14. Klisch, A., Atzberger, C.: Operational drought monitoring in Kenya using MODIS NDVI time series. *Remote Sens.* **8**, 267 (2016)
15. AEMET: Daily precipitation, maximum temperature and minimum temperature. Period 2000–2015 (2013)



Spatio-Temporal Clustering of Earthquakes Based on Average Magnitudes

Yuki Yamagishi¹(✉), Kazumi Saito², Kazuro Hirahara², and Naonori Ueda²

¹ Faculty of Informatics, Shizuoka Institute of Science and Technology,
Fukuroi, Japan

yamagishi.yuki@sist.ac.jp

² Center for Advanced Intelligence Project, RIKEN, Kyoto, Japan
{kazumi.saito,kazuro.hirahara,naonori.ueda}@riken.jp

Abstract. In this paper, we address the problem of automatically extracting several clusters consisting of spatio-temporally similar earthquakes whose average magnitudes are substantially different from the total average. For this purpose, we propose a new method consisting of two phases: tree construction and tree separation. In the former phase, we employ one of two different declustering algorithms called single-link and correlation-metric developed in the field of seismology, while in the later phase, we employ a variant of the change-point detection algorithm, developed in the field of data mining. In our empirical evaluation using earthquake catalog data covering the whole of Japan, we show that the proposed method employing the single-link algorithm can produce more desirable results for our purpose in terms of the improvement of weighted sums of variances and visualization results.

1 Introduction

Our research objective is to develop methods for analyzing huge earthquake catalogs as large-scale complex networks, where nodes (vertices) correspond to earthquakes, and links (edges) correspond to the interaction between them. Technically, we are not only interested in knowing what is happening now and how it develops in the future, but also we are interested in knowing what happened in the past and how it caused by some changes in the distribution of the information as studied in [10, 21]. Thus, it seems worth putting some effort into attempting to find empirical regularities and develop explanatory accounts of basic properties in these complex networks. Such attempts would be valuable for understanding some structures and trends, and inspiring us to lead to the discovery of new knowledge and insights underlying these interactions.

The clustering of earthquakes is important for many applications in seismology, including seismic activity modeling, and earthquake prediction. In this paper, for a given earthquake catalog, we addressed the problem of automatically extracting several clusters consisting of spatio-temporally similar earthquakes

whose average magnitudes are substantially different from the total one. Especially, we intended to produce one relatively large cluster and the other small clusters having substantially different average magnitudes from the total one. To this end, we propose a new method by uniquely combining some techniques developed in two different fields, i.e., declustering algorithms [2, 4, 6] in the field of seismology and a change-point detection algorithm [23, 24] in the field of data mining. In our empirical evaluation using an earthquake catalog covering the whole of Japan, it was confirmed that we could generally obtain the clustering results each of which consists of one relatively large cluster and the other small clusters having substantially different average magnitude from the total one.

The paper is organized as follows. We describe related work in Sect. 2 and give our problem setting and the proposed methods for clustering an earthquake catalog in Sect. 3. We report and discuss experimental results using a real catalog in Sect. 4 and conclude this paper and address the future work in Sect. 5.

2 Related Work

In this paper, we propose a new method by combining declustering algorithms and a change-point detection algorithm developed in the two different fields of seismology and data mining, respectively. Thus, we describe some existing studies relating to these algorithms below.

2.1 Declustering Algorithms

Seismicity declustering is the process of separating an earthquake catalog into foreshocks, mainshocks, and aftershocks, and several algorithms have been developed from various perspectives [20]. The window method is known as a simple way of identifying mainshocks and aftershocks. As a beginning of this method, the lengths and durations of windows were proposed by Knopoff and Gardner [7, 11]. After that, the alternative window parameter settings are proposed by Uhrhammer [22], and comparative experiments were conducted by Molchan and Dmitrieva [14]. Meanwhile, the algorithm of Reasenber [18] called as the cluster method assumed an interaction zone centered on each earthquake. This method based on the previous work of Savage [19], and Molchan and Dmitrieva [14] provide a condensed summary of the original paper of Reasenber. As an alternative to deterministic declustering methods above, ideas of probabilistic separation appeared in the investigation of Kagan et al. [9]. Zhuang et al. [26–28] suggested the stochastic declustering method also called stochastic reconstruction to bring such a probabilistic treatment into practice based on the ETAS (epidemic-type aftershock sequence) model [15, 16]. The generalization of stochastic declustering by Marsan and Lengline [12, 13] has no specific underlying model, and can therefore accept any (additive) seismicity model. In other studies, Frohlich and Davis [4, 6] proposed the single-link cluster analysis based on a space-time distance between two earthquakes, and Hainzl et al. [8] proposed the estimating background rate based on inter-event time distribution. Based on the inter-event

times, the method by Bottiglieri et al. [3] uses the coefficient of variation of the times, and Frohlich and Davis [5] proposed the ration method which also exploits the inter-event times but without examining their distribution. As another cluster analysis with links, Baiesi and Paczuski [2] proposed a simple space-time metric to correlate earthquakes with each other and Zaliapin et al. [25] further defined the rescaled distance and time.

Among these declustering algorithms, we focused on the single-link cluster analysis proposed by Frohlich and Davis [4,6] and the correlation metric proposed by Baiesi and Paczuski [2]. In our proposed method, we employ one of these two algorithms alternatively in the tree construction phase.

2.2 Change-Point Detection Algorithms

Our research aim is in some sense the same, in the spirit, with the work by Kleinberg [10] and Swan and Allan [21]. They noted a huge volume of the time-series data, tried to organize it, and extract structures behind it. This is done in a retrospective framework, i.e., assuming that there is a flood of abundant data already and there is a strong need to understand it. Kleinberg's work is motivated by the fact that the appearance of a topic in a document stream is signaled by a "burst of activity" and identifying its nested structure manifests itself as a summarization of the activities over a period of time, making it possible to analyze the underlying content much easier. Kleinberg's method used a hidden Markov model in which bursts appear naturally as state transitions, and successfully identified the hierarchical structure of e-mail messages. Swan and Allan's work is motivated by the need to organize a huge amount of information in an efficient way. They used a statistical model of feature occurrence over time based on hypothesis testing and successfully generated clusters of named entities and noun phrases that capture the information corresponding to major topics in the corpus and designed a way to nicely display the summary on the screen (Overview Timelines). We also follow the same retrospective approach, i.e., we are not predicting the future, but we are trying to understand the phenomena that happened in the past.

We are interested in detecting spatio-temporal changes in the magnitude of earthquakes. For this purpose, by defining a set of links with some declustering algorithm described earlier, we construct a spatio-temporal network (spanning tree), where the nodes correspond to the observed earthquakes. After that, in order to analyze the burst of activity in an earthquake catalog and attempt to present an overview map, we employ a variant of the change-point detection algorithm proposed by Yamagishi et al. [23,24].

3 Proposed Method

Let $\mathcal{D} = \{(\mathbf{x}_i, t_i, m_i) \mid 1 \leq i \leq N\}$ be a set of observed earthquakes, where \mathbf{x}_i , t_i and m_i stand for a location vector, time and magnitude of the observed earthquake i , respectively. Here, we assume that these earthquakes are in order

from oldest to most recent, i.e., $t_i < t_j$ if $i < j$. In this paper, from the observed dataset \mathcal{D} , we address the problem of automatically extracting several clusters consisting of spatio-temporally similar earthquakes whose average magnitudes are substantially different from the total one. In what follows, we describe some details of our proposed algorithm consisting of two phases: tree construction and tree separation.

3.1 Tree Construction Strategies

Among several seismicity declustering algorithms, we focus on two studies, i.e., the single-link cluster analysis proposed by Frohlich and Davis [4,6], and the correlation-metric proposed by Baiesi and Paczusi [2], which are also referred to as the SL and CM strategies, respectively. In our experiments described later, it is shown that we obtain quite different extraction results by employing either one of these two strategies.

In the single-link strategy, with respect to two earthquakes i and j , the spatio-temporal metric $d_{i,j}$ is defined as

$$d_{i,j} = \sqrt{\|\mathbf{x}_i - \mathbf{x}_j\|^2 + C^2(t_j - t_i)^2}. \tag{1}$$

It was found that a spatio-temporal scaling constant $C = 1$ km/day gives satisfactory results. Then, an earthquake j is regarded as the aftershock (child node) of i^{SL} if the metric $d_{i,j}$ is minimized, i.e., $i^{SL}(j) = \arg \min_{1 \leq i < j} d_{i,j}$. Then,

based on the single-link strategy, we can define a spanning tree, where the nodes correspond to the observed earthquakes, and the links are defined by $\mathcal{T}^{SL} = \{(i^{SL}(j), j) \mid 2 \leq j \leq N\}$.

In the correlation-metric strategy, with respect to two earthquakes i and j such that $i < j$, the spatio-temporal metric $n_{i,j}$ is defined as

$$n_{i,j} = (t_j - t_i) \|\mathbf{x}_i - \mathbf{x}_j\|^{d_f} 10^{-b m_i}. \tag{2}$$

Here d_f is the fractal dimension set to $d_f = 1.6$, and b the parameter of the Gutenberg-Richter law set to $b = 0.95$. Again, an earthquake j is regarded as the aftershock (child node) of $i^{CM}(j)$ if the metric $m_{i,j}$ is minimized, i.e., $i^{CM}(j) = \arg \min_{1 \leq i < j} n_{i,j}$. Then, based on the correlation-metric strategy, we can

define a spanning tree, where the nodes correspond to the observed earthquakes, and the links are defined by $\mathcal{T}^{CM} = \{(i^{CM}(j), j) \mid 2 \leq j \leq N\}$.

3.2 Tree Separation Algorithm

Let $\mathcal{R} \subset \mathcal{T}$ be a subset of tree links constructed by either the SL and CM strategies. Here note that when $|\mathcal{R}| = G - 1$, by removing all the links in \mathcal{R} from \mathcal{T} , we can separate the tree into G connected components. Then, the original set of observed earthquakes which correspond to nodes of the tree is also divided

into G clusters as $\{\mathcal{N}_g \mid 1 \leq g \leq G\}$, where $\mathcal{N}_1 \cup \dots \cup \mathcal{N}_G = \{1, \dots, N\}$. Now, by denoting the average magnitude of cluster \mathcal{N}_g as

$$\mu_g = \frac{1}{|\mathcal{N}_g|} \sum_{i \in \mathcal{N}_g} m_i, \quad (3)$$

we can derive our objective function to be minimized as follows:

$$f(\mathcal{R}) = \sum_{g=1}^G \frac{|\mathcal{N}_g|}{N} \frac{1}{|\mathcal{N}_g|} \sum_{i \in \mathcal{N}_g} (m_i - \mu_g)^2 = \frac{1}{N} \sum_{g=1}^G \sum_{i \in \mathcal{N}_g} (m_i - \mu_g)^2. \quad (4)$$

Note that we employed the definition of sample variance in this paper. Intuitively, we intend to produce one relatively large cluster and the other small clusters having substantially different (typically large) average magnitudes from the total one. In fact, since the distribution of magnitudes in a catalog reasonably obeys the Gutenberg-Richter law (exponential distribution), i.e., the magnitudes of most earthquakes are relatively small, it is naturally expected that we can improve the objective function by separating clusters of spatio-temporally similar earthquakes with relatively large magnitudes. Here note that this objective function can be interpreted as a weighted sum of variances.

In order to compute the resultant set of separation links \mathcal{R} , we employ a variant of the change-point detection algorithm proposed by Yamagishi et al. [23, 24]. Namely, from the observed dataset \mathcal{D} , the tree \mathcal{T} constructed by either the SL and CM strategies, and a given number of clusters G , our algorithm computes \mathcal{R} as follows:

- Step 1.** Initialize $g \leftarrow 1$ and $\mathcal{R}_0 \leftarrow \emptyset$.
- Step 2.** Compute $e_g \leftarrow \arg \min_{e \in \mathcal{T}} \{f(\mathcal{R}_{g-1} \cup \{e\})\}$, and update $\mathcal{R}_g \leftarrow \mathcal{R}_{g-1} \cup \{e_g\}$.
- Step 3.** Set $g \leftarrow g + 1$ and then return to Step 2 if $g < G - 1$; otherwise set $g \leftarrow 1$ and $h \leftarrow 0$,
- Step 4.** Compute $e'_g = \arg \min_{e \in \mathcal{T}} \{f(\mathcal{R}_G \setminus \{e_g\} \cup \{e\})\}$, and update $\mathcal{R}_G \leftarrow \mathcal{R}_G \setminus \{e_g\} \cup \{e'_g\}$ and then $h \leftarrow 0$ if $e'_g \neq e_g$; otherwise set $h \leftarrow h + 1$,
- Step 5.** Output \mathcal{R}_{G-1} and then terminate if $h = G - 1$; otherwise set $g \leftarrow (g \bmod (G - 1)) + 1$ and then return to Step 4.

More specifically, after initializing the variables in Step 1, we compute the optimal g -th link in e_g by fixing the already selected set of $(g - 1)$ links in \mathcal{R}_{g-1} and add it to \mathcal{R}_{g-1} as shown in Step 2. We repeat this procedure from $g = 1$ to $G - 1$ as shown in Step 3. After that, we start with the solution obtained as \mathcal{R}_{G-1} , pick up a link e_g from the already selected links, fix the rest $\mathcal{R}_{G-1} \setminus \{e_g\}$ and compute the better link e'_g of e_g as shown in Step 4, where $\cdot \setminus \cdot$ represents set difference. We repeat this from $g = 1$ to $G - 1$. If no replacement is possible for all g , i.e. $e'_g = e_g$ for all $g \in \{1, \dots, G - 1\}$, then no better solution is expected and the iteration stops, as shown in Step 5. Here, it is not guaranteed that the above algorithm theoretically produces the optimal result, but it is confirmed that the algorithm always computes the optimal or near-optimal solutions in our empirical evaluation [23, 24].

4 Experimental Evaluation

By using an earthquake catalog which contains source parameters determined by Japan Meteorological Agency¹ in the whole of Japan Islands, we generated two original datasets. Namely, by setting the minimum magnitude and the maximum depth as $M_{\min} = 3.0$ and $D_{\max} = 100$ km, respectively, we selected $N = 104,343$ earthquakes during the period from Oct. 01, 1997 to Dec. 31, 2016 as dataset A, while by setting $M_{\min} = 4.0$ and $D_{\max} = 100$ km, we selected $N = 27,728$ earthquakes during the period from Oct. 01, 1977 to Dec. 31, 2016 as dataset B.

4.1 Quantitative Evaluation

First, we evaluate the performance of the proposed method employing our different tree construction strategies, i.e., single-link and correlation-metric. Figure 1 shows the experimental results of the datasets A and B, which are depicted in Figs. 1a and b, where the horizontal and vertical axes stand for the number of clusters varied from $G = 1$ to 8 and the objective function value defined in Eq. (4), respectively. Note that for each of Figs. 1a and b, the value at $G = 1$ is nothing more than the total variance of each dataset. From these experimental results, we can see that in the case of employing the single-link strategy, the objective function values interpreted as the weighted sums of variances become much smaller in comparison to those of employing the correlation-metric strategy. This suggests that the proposed method employing the single-link strategy can produce more desirable results for our purpose.

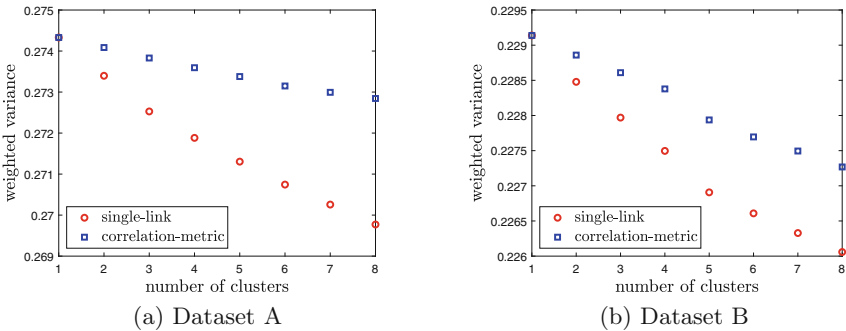


Fig. 1. Results of performance evaluation

Next, we evaluate the similarity of the results with different numbers of clusters obtained by the proposed method. In what follows, we only show our experimental results using the dataset A due to a space limitation, but reasonably similar results have been obtained for the dataset B. For this purpose, we employ the Rand index [17] for evaluating two different clustering results

¹ <https://www.data.jma.go.jp/svd/eqev/data/bulletin/hypo.html>.

denoted by $\mathcal{G} = \{\mathcal{N}_g \mid 1 \leq g \leq G\}$ and $\mathcal{H} = \{\mathcal{N}'_h \mid 1 \leq h \leq H\}$, where $G \neq H$ in general and $\mathcal{N}_1 \cup \dots \cup \mathcal{N}_G = \mathcal{N}'_1 \cup \dots \cup \mathcal{N}'_H = \{1, \dots, N\}$. More specifically, for an earthquake i , let $g(i) \in \{1, \dots, G\}$ and $h(i) \in \{1, \dots, H\}$ be the cluster number indicator functions of \mathcal{G} and \mathcal{H} , respectively. Then, we can compute the Rand index $I(\mathcal{G}, \mathcal{H})$ as

$$I(\mathcal{G}, \mathcal{H}) = \frac{|\{i, j \in \mathcal{M} \mid (g(i) = g(j) \wedge h(i) = h(j)) \vee (g(i) \neq g(j) \wedge h(i) \neq h(j))\}|}{|\mathcal{M}|}, \quad (5)$$

where $\mathcal{M} = \{i, j \in \{1, \dots, N\} \mid i \neq j\}$. Figure 2 shows the similarity matrices consisting of the Rand index by varying the number of clusters from $G = 2$ to 8, where Figs. 2a and b are those of the proposed method employing the single-link and correlation-metric strategies. From these experimental results, we can see that in the case of employing the single-link strategy, there exist three types of similar results, i.e., $2 \leq G \leq 4$, $5 \leq G \leq 7$ and $G = 8$, but almost a single type of similar results except for $G = 2$ in the case of employing the correlation-metric strategy. Namely, we can expect to obtain several types of results by varying G in the case of the single-link strategy.

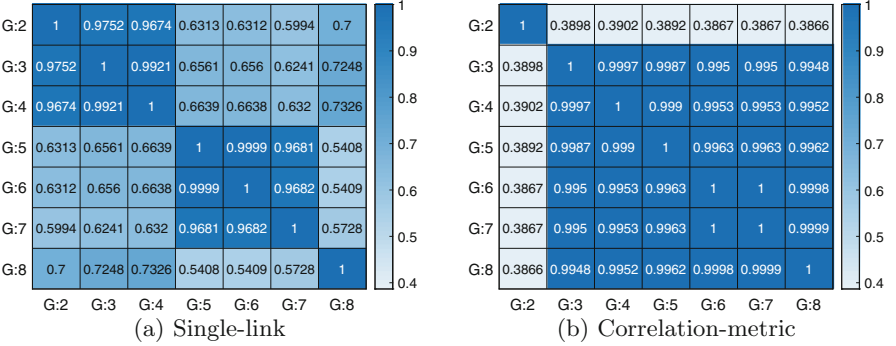


Fig. 2. Results of similarity evaluation

4.2 Visual Evaluation

Finally, we visually evaluate the obtained results employing the different tree construction strategies by focusing on the dataset A. To this end, we transform the average magnitude in each cluster \mathcal{N}_g denoted by μ_g , into the corresponding b -value denoted as b_g [1], i.e.,

$$b_g = \frac{\log_{10} e}{\mu_g - M_{\min}}, \quad (6)$$

where e stands for Napier's constant (Euler's number) and recall that the minimum magnitude in the dataset A was set to $M_{\min} = 3.0$. Here the average magnitude in the dataset A is around 3.50, and the corresponding b -value is

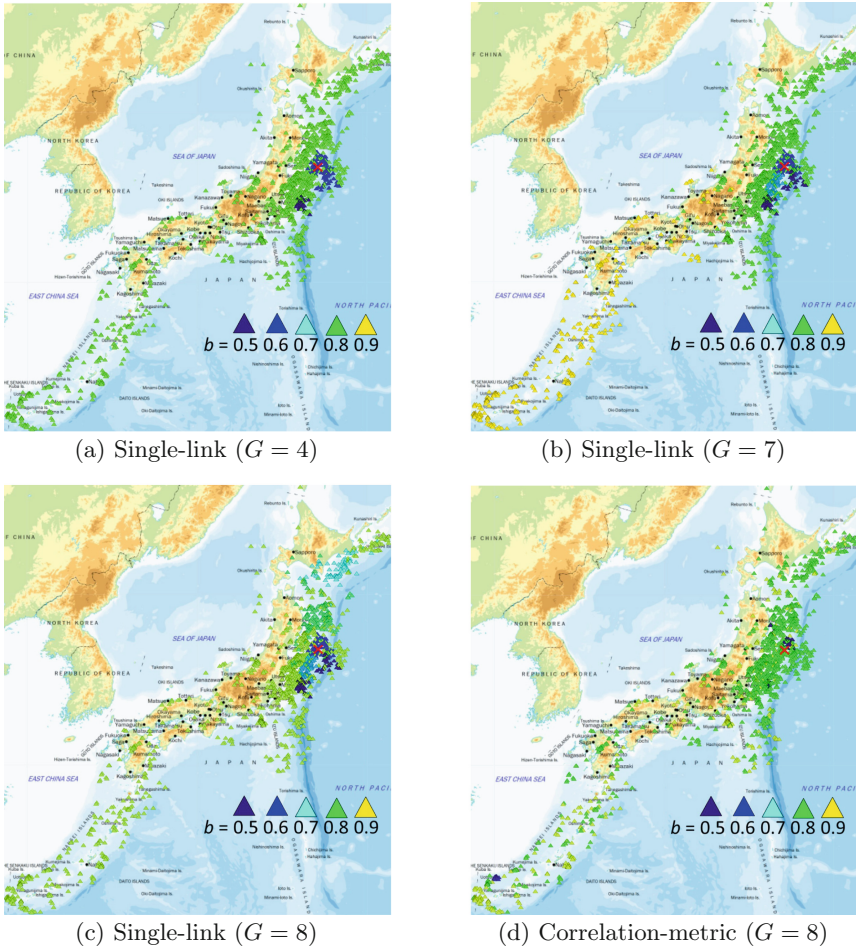


Fig. 3. Results of visual evaluation (the largest magnitude earthquake in Japan is indicated by red x)

approximately amount to 0.86. Figure 3 shows our visualization results whose numbers of clusters are 4, 7, and 8 for the single-link strategy, and 8 for the correlation-metric strategy, where these results are selected according to the similarity matrices shown in Fig. 2. Also note that by selecting earthquakes whose magnitudes are greater than or equal to 5.0, we plotted each of them as a triangle with a color shown in Fig. 3 according to its cluster's corresponding b -value.

From these results, as expected, we could generally obtain the clustering results each of which consists of one relatively large cluster and the other small clusters having substantially different average magnitude from the total average. As for the comparison between the two different strategies, single-link and correlation-metric, shown in Figs. 3c and d, respectively, by employing the for-

mer strategy, we could obtain clearly visible clusters having substantially large average magnitudes around the region where the 2011 Tohoku earthquake with $M_w = 9.0$ (indicated by red x in the figures), the largest magnitude in Japan, occurred. On the other hand, as for comparison among the different numbers of clusters in case of employing the single-link strategy, shown in Figs. 3a, b and c, we could obtain somehow different types of clustering results, which might help to analyze the dataset from multiple viewpoints. In short, in our empirical evaluation, we can confirm that the proposed method employing the single-link strategy can produce more desirable results for our purpose.

5 Conclusion

In this paper, for a given dataset of observed earthquakes, we addressed the problem of automatically extracting several clusters consisting of spatio-temporally similar earthquakes whose magnitudes are substantially different from the total average. Especially, we intended to produce one relatively large cluster and the other small clusters having substantially different average magnitudes from the total one. For this purpose, we proposed a new method consisting of two phases. In the former tree construction phase, we employ one of two different declustering algorithms called single-link and correlation-metric developed in the field of seismology, while in the later tree separation phase, we employ a variant of the change detection algorithm, developed in the field of data mining. In our empirical evaluation using earthquake catalog data covering the whole of Japan, it was confirmed that we could generally obtain the clustering results each of which consists of one relatively large cluster and the other small clusters having substantially different average magnitude from the total one. Moreover, we showed that the proposed method employing the single-link strategy can produce more desirable results, in terms of the improvement of weighted sums of variances and visualization results. As a future task, we plan to conduct more experiments to see that our clustering method can provide new findings on the earthquake statistics, the underlying earthquake dynamics, and so on, by producing one relatively large cluster and the other small clusters having substantially different average magnitudes from the total one. Further theoretical studies to find the optimal number of clusters are also future works.

References

1. Aki, K.: Maximum likelihood estimate of bin the formula $\log(N) = a - bM$ and its confidence limits. *Bull. Earthq. Res. Inst.* **43**, 237–239 (1965)
2. Baiesi, M., Paczuski, M.: Scale-free networks of earthquakes and aftershocks. *Phys. Rev. E, Stat. Nonlinear Soft Matter Phys.* **69**, 066106 (2004)
3. Bottiglieri, M., Lippiello, E., Godano, C., De Arcangelis, L.: Identification and spatiotemporal organization of aftershocks. *J. Geophys. Res.* **114** (2009)
4. Davis, S.D., Frohlich, C.: Single-link cluster analysis, synthetic earthquake catalogues, and aftershock identification. *Geophys. J. Int.* **104**(2), 289–306 (1991)

5. Frohlich, C., Davis, S.: Identification of aftershocks of deep earthquakes by a new ratios method. *Geophys. Res. Lett.* **12**, 713–716 (1985)
6. Frohlich, C., Davis, S.D.: Single-link cluster analysis as a method to evaluate spatial and temporal properties of earthquake catalogues. *Geophys. J. Int.* **100**(1), 19–32 (1990)
7. Gardner, J.K., Knopoff, L.: Is the sequence of earthquakes in Southern California, with aftershocks removed, Poissonian? *Bull. Seismol. Soc. Am.* **64**(5), 1363–1367 (1974)
8. Hainzl, S., Scherbaum, F., Beauval, C.: Estimating background activity based on interevent-time distribution. *Bull. Seismol. Soc. Am.* **96**, 313–320 (2006)
9. Kagan, Y., Jackson, D.: Long-term earthquake clustering. *Geophys. J. Int.* **104**, 117–133 (1991)
10. Kleinberg, J.: Bursty and hierarchical structure in streams. In: *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002)*, pp. 91–101 (2002)
11. Knopoff, L., Gardner, J.K.: Higher seismic activity during local night on the raw worldwide earthquake catalogue. *Geophys. J. Int.* **28**, 311–313 (1972)
12. Marsan, D., Lengliné, O.: Extending earthquakes' reach through cascading. *Science* **319**(5866), 1076–1079 (2008)
13. Marsan, D., Lengliné, O.: A new estimation of the decay of after shock density with distance to the mainshock. *J. Geophys. Res.: Solid Earth*, **115** (2010)
14. Molchan, G.M., Dmitrieva, O.E.: Aftershock identification: methods and new approaches. *Geophys. J. Int.* **109**, 501–516 (1992)
15. Ogata, Y.: Statistical models for earthquake occurrences and residual analysis for point processes. *J. Am. Stat. Assoc.* **83**(401), 9–27 (1988)
16. Ogata, Y.: Space-time point-process models for earthquake occurrences. *Ann. Inst. Stat. Math.* **50**, 379–402 (1998)
17. Rand, W.M.: Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **66**(336), 846–850 (1971)
18. Reasenber, P.: Second-order moment of Central California seismicity, 1969–1982. *J. Geophys. Res.* **90**, 5479–5495 (1985)
19. Savage, W.U.: Microearthquake clustering near Fairview Peak, Nevada, and in the Nevada seismic zone. *J. Geophys. Res.* **77**, 7049–7056 (1972)
20. van Stiphout, T., Zhuang, J., Marsan, D.: Seismicity declustering. *Community Online Resource for Statistical Seismicity Analysis* (2012)
21. Swan, R., Allan, J.: Automatic generation of overview timelines. In: *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000)*, pp. 49–56 (2000)
22. Uhrhammer, R.A.: Characteristics of Northern and Central California seismicity. *Earthq. Notes* **57**(1), 21–37 (1986)
23. Yamagishi, Y., Okubo, S., Saito, K., Ohara, K., Kimura, M., Motoda, H.: A method to divide stream data of scores over review sites. In: *PRICAI 2014: Trends in Artificial Intelligence - 13th Pacific Rim International Conference on Artificial Intelligence. Lecture Notes in Computer Science*, vol. 8862, pp. 913–919. Springer (2014)
24. Yamagishi, Y., Saito, K.: Visualizing switching regimes based on multinomial distribution in buzz marketing sites. In: *Foundations of Intelligent Systems - 23rd International Symposium, ISMIS 2017. Lecture Notes in Computer Science*, vol. 10352, pp. 385–395. Springer (2017)
25. Zaliapin, I., Gabriellov, A., Keilis-Borok, V., Wong, H.: Clustering analysis of seismicity and aftershock identification. *Phys. Rev. Lett.* **101**(1), 1–4 (2008)

26. Zhuang, J.: Multi-dimensional second-order residual analysis of space-time point processes and its applications in modelling earthquake data. *J. R. Stat. Soc.* **68**(4), 635–653 (2006)
27. Zhuang, J., Ogata, Y., Vere-Jones, D.: Stochastic declustering of space-time earthquake occurrences. *J. Am. Stat. Assoc.* **97**(458), 369–380 (2002)
28. Zhuang, J., Ogata, Y., Vere-Jones, D.: Analyzing earthquake clustering features by using stochastic reconstruction. *J. Geophys. Res.* **109** (2004)

Information Spreading in Social Media



Analyzing the Robustness of a Comprehensive Trust-Based Model for Online Social Networks Against Privacy Attacks

Nadav Voloch¹ (✉), Ehud Gudes¹, and Nurit Gal-Oz²

¹ Ben-Gurion University of the Negev, P.O.B. 653, 8410501 Beer-Sheva, Israel
voloch@post.bgu.ac.il

² Sapir Academic College, 79165 Sderot, Israel

Abstract. Security and privacy have been major concerns of Online Social Networks (OSN). Individual users as well as organizations utilize OSNs, such as Facebook, Twitter, and LinkedIn, to share information with other users within their networks. While sharing information, users are not always aware of the fact that an innocent action on their post by a direct friend such as a comment or a share may turn the post transparent to someone outside their network.

In previous work we have devised a comprehensive Trust-based model that combines Role based Access Control for the direct circle of friends and Flow Control for the friends' networks. In this paper we reinforce this model by analyzing its strength in terms of OSN features. We simulate attack scenarios carried out by a community of malicious users that attempt to fake the OSN features of the model. We analyze the attack of an alleged trustworthy clique of adversaries and show the futility of such an attack, due to the strength of the model's parameters and combination of Trust, Access Control and Flow Control. We also demonstrate the robustness of the model when facing an optimized attack, which carefully selects the best network nodes to compromise, as determined by the minimal vertex cover algorithm.

Keywords: Social networks privacy · Access control · Flow control · Network attacks

1 Introduction

The rapid growth of Online Social Networks (OSN) and their increasing popularity in the past decade as major communication channels, have raised some new shapes of security and privacy concerns. In our previous work, we have created a privacy model that is composed of three main phases addressing three of its major aspects: trust, role-based access control [1, 2] and information flow, by creating an Information Flow-Control model for adversary detection [3], or a trustworthy network [4]. We represent a social network as an undirected graph, where nodes are the OSN users, and edges represent relations between them such as friendship relations. An Ego node (or Ego user) is an individual focal node, representing a user whose information flow we aim

to control. An Ego node along with its adjacent nodes are denoted Ego network. Our comprehensive Trust-based model uses Access Control for the direct friends of the Ego-user, and Information Flow Control for the users that are in a further distance. We use OSN parameters, such as total number of friends, age of user account, and friendship duration to characterize the quality of the network connections as we explain in Sect. 4 of this paper. The robustness of this model is the key objective of this paper. Several attacks on private information in Social Networks have been described in [5]. A common type of attack in OSN aims at a specific user or network and attempts to access or act on its information e.g., spread false data or spam for different purposes. Trust based systems must deal with attacks, in which malicious users initially behave properly to gain a positive reputation but then start to misbehave and inflict damage on the community. In this paper we show the robustness of our model and focus on the latter type of attack, where a user, or its network is the target of an attack initiated by malicious users. The main scenarios we simulate include a community of spammers whose profiles conform with the OSN attributes that constitute the Trust aspect of the model. We use a graph algorithm (minimal vertex cover) to select an optimized set of candidate nodes to compromise, and show that even in this case, such an attack is futile. The rest of this paper is structured as follows: Sect. 2 discusses the background for our work and references related papers; Sect. 3 provides a brief overview of the Trust model; Sect. 4 discusses the attack scenarios on the model and Sect. 5 presents the experimental evaluation of the attack scenario based on preliminary evaluations of the properties conducted in our previous research. In Sect. 6 we conclude the paper and discuss further research directions.

2 Background and Related Work

The model we discuss in this paper was presented in previous work [1–4], and as briefly described in Sect. 3, it combines Access Control, Information Flow Control and Trust. The main Access Control model used in OSN is Role-Based Access Control (RBAC) that has many versions, as presented in [6], and limits access by creating user-role assignments. The user must have a role that has permission to access that resource. The most prominent advantage of this method is that permissions are not assigned directly to users but to roles, making it much easier to manage the access control of a single user, since it must only be assigned the right role. An addition to this model is the Trust factor [7], which is based on the users' interactions history. However, for new connections, there is no way to evaluate trust. In our model we overcome this limitation by using the independent user attributes to estimate trust.

Collusion attacks, in which a group of malicious users act together with strong trust relations between them to manipulate the system and gain high reputation and then cause damage in the Social Networks are described in [8–10]. Our simulated attacks on the model differ from Collusion attacks on reputation systems, such as the one described in [8], as we attack the privacy of the Ego-user, based on the Trust criteria established in our above mentioned model. The problem of such attacks on trust criteria is presented in [11], where a reputation Lag attack is described as a formal model capturing the core properties of the attack, in which the reputation of a user fails to reflect their behavior

due to a delay and a malicious user exploits this delay for a personal gain. Attacks on social networks are presented in relatively early papers such as [12], where a conceptual framework of a Social Honeypot is described for uncovering social spammers who target online communities. The idea of creating Social Honeypots is also developed in [13] where the honeypot profiles were assimilated into an organizational social network. The honeypot then received suspicious friend requests and mail messages that revealed basic indications of a potential forthcoming attack. An interesting form of attack, that is related to our presented attack scenarios is the “friend-in-the-middle” attack [14], in which a legitimate friend in the social network is used as a gateway for spammers that harvest social data. This data can then be exploited for large-scale attacks such as context-aware spam and social-phishing. The network used specifically in this attack scenario is Facebook. Our Vertex-Cover algorithm can be looked at as a generalization of the “Friend-in-the-middle” attack.

3 The Comprehensive Trust-Based Model

The model we have presented in previous work [1–4] is composed of three main phases addressing three of its major aspects: trust, role-based access control and information flow. In the First phase, the Trust phase, we assign trust values on the edges connecting direct friends to the Ego node in their different roles, e.g., Family, Colleagues etc. In the second phase, the Role Based Access Control phase, we remove direct friends that do not have the minimal trust values required to grant a specific permission to their roles. A cascade removal is carried out in their Ego networks as well.

After this removal, the remaining user nodes and their edges are also assigned with trust values. In the third and last phase, the Information Flow phase, we remove from the graph edges and nodes that are not directly connected to the Ego-user to construct a privacy preserving trusted network. To calculate trust values in the first phase we use a set of OSN parameters carefully selected based on previous research (specifically [1, 2, 4]). We divide these parameters to *connection attributes* which relate to edges and to *user attributes* which relate to nodes. In this work we refer to four of these attributes. Two connection attributes: Friendship Duration (*FD*) and Mutual Friends (*MF*) and another two user attributes: Total number of Friends (*TF*) and Age of User Account (*AUA*). A Trust value ranges between 0 and 1 to reflect the probability of sharing information with a certain user: 0 represents total restriction, and 1 represents definite sharing willingness. The threshold values are denoted here as $T^{property}$ (e.g. for the *TF* attributes the threshold value is T^{TF}) and their experimental values, achieved in our previous research mentioned above are presented in the Evaluation part of this paper. We define the User Trust Value (*UTV*), as the weighted average of these properties, taking into consideration the different weights (w_i) that were assessed by experimental results in [1] and [2] for the significance (weight) of every attribute-factor. The calculation of a certain property value ($p_{property}$) is done by these thresholds and is as follows:

$$P_{property} = \begin{cases} \frac{property}{T^{property}} & (property < T^{property}), \\ 1 & (property \geq T^{property}). \end{cases} \quad (1)$$

The User Trust Value (*UTV*) is calculated as follows, where $|p|$ denotes the number of attributes and $\langle w \rangle$ denotes the average of their weights:

$$UTV = \langle w_i p_i \rangle = \frac{\sum_{i=1}^{|p|} w_i p_i}{\langle w \rangle |p|} \tag{2}$$

Minimal Trust Value (*MTV*) denotes the threshold value for determining whether to grant a certain access to a data instance. It is calculated in this model as the average of *UTVs* within the Ego Network.

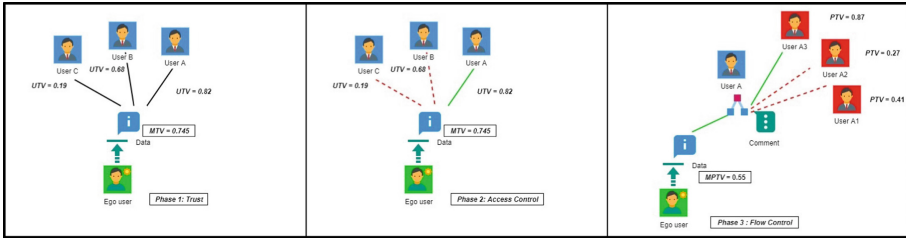


Fig. 1. The model’s phases for creating a trustworthy network

These thresholds can be configured per role and per permission according to the OSN administration policy, or according to user-preferences if such exist.

The three-phases model described above and presented in Fig. 1. generates a trustworthy network of users with which the Ego-user can safely share information. In Phase 3 we use the measure Minimal Path Trust Value (*MPTV*) that is presented in [3], and is the threshold value for *PTV* (Path Trust Value), that is computed as the cumulative Trust values of the edges and nodes in the path from the Ego user to the user being checked in the network.

4 Analysis of Attack Scenarios on the Model

4.1 Attack Definitions and Scenario

To examine the vulnerability of our Trust-based comprehensive model we question the strength of the Trust attributes that are used to determine the levels of trustworthiness in the Ego-user’s network. To gain a high user trust level (*UTV*) an attacker must fake all the values of the relevant attributes required to build this trust level. In this section we consider possible attacks on these attributes and analyze the feasibility of such attacks. To create a fake user that appears genuine an attacker should make sure that the user is connected to other users. An attack on the model is the creation of a set of fake users such that each fake user has its own ego network. The success of an attack depends on the network of the fake users so usually it would be a collaborating network of the fake users so usually, it would be a collaborating network of fake users. To formalize this attack, we provide the following definition:

Definition 1- An attack is a tuple of the form $\langle G, T^{TF}, T^{AUA}, G^{spm}, t_{spm} \rangle$ where:

T^{TF} is the *Total Friends* threshold value of the Ego user network

T^{AUA} is the *Age of User Account* threshold value of the Ego user network

G – is the graph of the Ego user that is under attack.

$G^{spm} (V^{spm}, E^{spm})$ – the spammer graph that is created in the attack.

t_{spm} - the elapsed time before the attack can take place

The result of the attack is denoted:

$G^\psi = G \cup G^{spm}$ – the spammed graph after the attack

For an attack to take place, the model major trust attributes must be faked: *Mutual Friends (MF)*, *Total Friends (TF)*, *Age of User Account (AUA)* and *Friendship Duration (FD)*. We divide these attributes into two groups: attributes representing quantities (*MF* and *TF*), and attributes representing durations (*AUA* and *FD*). Quantities imply that a user is well connected and a user that has enough mutual friends with others demonstrates human circles of relations within an OSN (family, work, neighborhood, etc.). Duration attributes represent the steadiness of the profile, as genuine users usually create their profile once. To fake a user attribute such as *MF* or *TF* an adversary must connect the fake user profile to other profiles in the network, genuine or not. The minimal number of fake users to be created must exceed the threshold of every attribute. To impersonate to a real user network, an attack must consist of a network of trustworthy users, that need to adhere to all the model's properties. We consider the extreme scenario of spammers that are only friends with each other, making the *MF* property similar as possible to the *TF* property. This attack simulates a closed spammer network $G^{spm} (V^{spm}, E^{spm})$ that is a clique. In this type of attack the *MF* attribute is correctly faked, since all the users are connected to each other. As all the nodes are connected in the spammer clique the size of the spammer graph must be at least:

$$|V^{spm}| \geq T^{TF} \quad (3)$$

For the duration attributes, *AUA* and *FD*, we also consider the extreme scenario of spammers that are only friends with each other, making the *FD* property similar as possible to the *AUA* property. These properties must also hold for all the users in the spammer's network. This is specifically hard due to OSN policies that require a reasonable duration for a user account to be considered a genuine one. This means that before the attack can take place the elapsed time should be:

$$t_{spm} \geq T^{AUA} \quad (4)$$

This attack process is shown in Fig. 2, where the two properties are created.

The creation of a spammer network for malicious purposes, is detailed in [15], where the following attack is described: a malicious user that creates a set of false identities and uses them to communicate with a large, random set of innocent users (Random Link Attack-RLA). The research shows and proves that this is in fact an NP-complete problem. Practically it means that this kind of attack, carried out naively without heuristics, is very

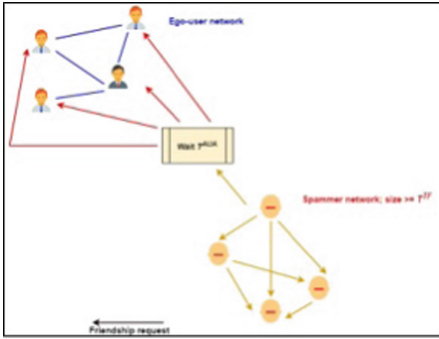


Fig. 2. Attack of a spammer network on the Ego

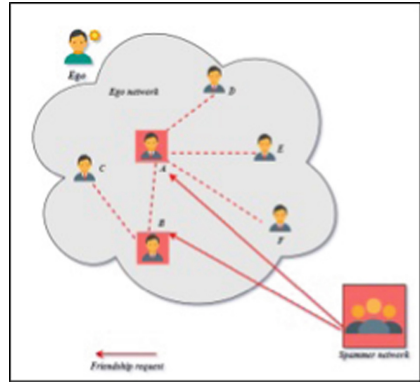


Fig. 3. Minimal vertex cover: optimized attack

hard to perform. We extend this form of basic attack one step further as we take into consideration the attributes of these nodes, making the attack even more difficult to implement. To perform an efficient attack, we need to assume that some of the requests of the spammer network will be denied or blocked by the OSN administration, thus the attack has to involve as many friends as possible from the Ego network. The robustness of our model is derived from its resilience to these attacks in term of actual OSN size - the bigger the network is, the harder it is to fake the attributes of the model. We define four types of attack, based on their strength and complexity:

A Regular Attack: A Blackbox attack that does not include preliminary knowledge on the Ego-user network. In this attack the spammer network tries to connect to k direct friends of the Ego network, where $1 \leq k \leq \frac{T^{TF}}{2}$.

In this attack, the number of friend requests to be made, is the number of edges from the spammed network, that is $|E^{spm}|$. Since it is a clique, $|E^{spm}| = \frac{|V^{spm}|(|V^{spm}| - 1)}{2}$, and the size of the connected network is:

$$|E^\psi| = \frac{|V^{spm}|(|V^{spm}| - 1)}{2} + k \cdot T^{TF} = \frac{T^{TF}(T^{TF} - 1)}{2} + k \cdot T^{TF} \tag{5}$$

Since usually MF is much smaller than TF, even if many of the friend requests are denied, MF will be fulfilled. This attack has the extreme case where MF = TF.

A Strong Attack: Another Blackbox attack, in which the spammer network tries to connect to all direct friends of the Ego user. In this case, the number of edges from the spammed network, representing the number of friend requests to be made is:

$$|E^\psi| = \frac{|V^{spm}|(|V^{spm}| - 1)}{2} + (T^{TF})^2 = \frac{T^{TF}(T^{TF} - 1)}{2} + (T^{TF})^2 \tag{6}$$

A Very Strong Attack: This is a knowledge-based Whitebox attack, that includes the pre-requisite of being familiar with the Ego-network structure.

In this attack the spammer network tries to connect to all the friends within a distance d from the Ego user. In this case, the number of friend requests that should be made, which are the number of edges from the spammed network, is:

$$|E^\psi| = \frac{|V^{spm}|(|V^{spm}| - 1)}{2} + (T^{TF})^d = \frac{T^{TF}(T^{TF} - 1)}{2} + (T^{TF})^d \quad (7)$$

An Optimized Very Strong Attack: An attack that uses an optimization algorithm to conduct an efficient attack. In the next sub section, we describe a minimization heuristic that a smart spammer would perform, but as we show this problem is still NP-complete.

The complexity of the problem of creating a fake friends' network becomes harder as the attack strength grows, and therefore it is not viable in terms of OSN sizes.

4.2 Optimizing the Attack: Minimizing the Connections of Fake Users by Reduction from Minimum Vertex Cover

An attempt of a spammer's network to reach out to the entire Ego network could create an anomalous amount of action in the OSN, which may raise the suspicion of the OSN administration or community. Certain techniques for minimizing this amount of activity may involve graph algorithms to allow the attacker an efficient connection to several nodes in the Ego-users graph instead of connecting to the entire Ego network. In graph theory, a vertex cover of a graph is a set of vertices such that each edge in the graph is incident to at least one vertex of the set. Formally, a vertex cover V' of an undirected graph $G = (V, E)$ is a subset of V such that $uv \in E \wedge (u \in V' \vee v \in V')$.

It is a set of vertices V' where every edge has at least one endpoint in the vertex cover V' . Such a set is said to cover the edges of G . The problem of finding a minimum vertex cover in a graph is an optimization problem [16]. We assume that a sophisticated attacker would create fake attributes only on the vertices (users) in V' , which are in the minimum vertex cover, enabling the attacker to control all of the connections with a minimal number of users, which require minimal effort in terms of actions required for the creation of the attack graph. To reduce the problem to the spammer attack we define the cost $c^\psi(v) \geq 0$ as the number of actions required for the creation of G^ψ and formulate as follows: *minimize* $\sum_{v \in V} c^\psi(v)x_v$ (minimize the total number of actions)

subject to $x_v + x_{v^{spm}} \geq 1$ for all $\{v^{spm}, v\} \in E^{spm}$ (cover every edge of the connected spammed subgraph that connects a spammer node with a friend node)
 $x_v \in \{0, 1\}$ for all $v \in V$ (every vertex is either in the vertex cover or not)
 $G^\psi \leftarrow G^{spm} \cup V'$ (the connection of the spammer network is to the vertex cover)

An example of such a minimal vertex cover is seen in Fig. 3. $V' = \{A, B\}$ is a vertex cover, since all of the edges are connected either to A or B . There are three major reasons to the futility of such an attack: first, the problem of finding the minimal vertex cover is NP-complete ([16]). Second, the networks of the allotted users in V' , remain very big, and must be created with fake attributes as presented in the previous subsection. Finally, after the creation of the spammer network, the attack is being delayed by t_{spm} .

This delay in time could be very significant in terms of the OSN structure: as time goes by, properties change, users are added and removed, and the network can be different from its preliminary status. The changes of the network create a difficulty for an attack that was pre-ordained to the original network and might not be relevant after the delay of T^{AUA} . The full attack is described in Algorithm 3.

Algorithm 3. <i>SpammerCommunityMinimalAttackOnOSN</i>
Input: Total Friends threshold T^{TF} , Age of User Account threshold T^{AUA} , Graph G , Spammer Vertex v_{spm_0} ; Output: Spammed Graph G^ψ
For $i=1$ to T^{TF} $v_{spm_i} \in V^{spm}$ // creating T^{TF} fake users Graph $G^{spm} \leftarrow \{V^{spm}, E^{spm}\}$; // creating a spammer network Wait (T^{AUA}) // the threshold time must pass to authenticate the AUA attribute $V' \leftarrow \text{minimalVertexCover}(G)$ For each v in V' and e in E ; $0 \leq i \leq V' $ $e_i \leftarrow \{v_{spm_i}, v_i\}$ // spammer connects to minimalVertexCover of Ego network $G^\psi \leftarrow G \cup G^{spm}$ return G^ψ

5 Evaluation

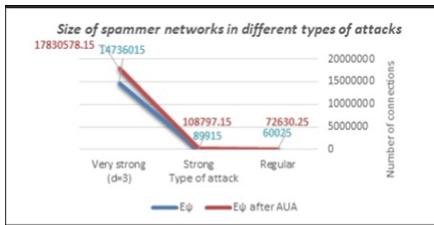
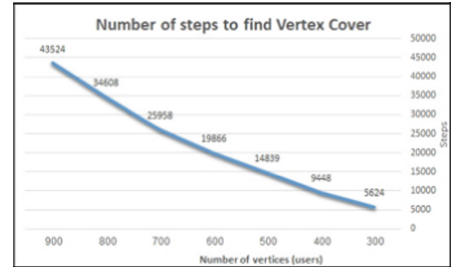
The experimental evaluation estimates the attacker effort in terms of the size of the spammer network that is required for a successful attack to take place.

For the OSN attributes threshold we have used the results obtained from our previous research [2] as presented in Table 1. These threshold values were obtained from a survey of 282 OSN users that were asked for the importance of various attributes in their decisions to grant various permissions to their private data. The MTV value was calculated based on UTV values (Eq. 2) using a real OSN dataset that included the attributes of 162 users of an Ego-network [2]. To calculate the sizes of spammer networks we use the experimental results of the thresholds values T^{TF} and T^{AUA} (Table 1) and use Eqs. 3 and 4 presented above. We can see that the basic T^{TF} is 245 for $d = 1$, and T^{AUA} is 24 months. The size of the spammer network in terms of edges being created is expressed by $|E^{spm}|$, and as described above, since it is a clique, $|E^{spm}| = \frac{|V^{spm}|(|V^{spm}|-1)}{2}$. The result graph and values are shown in Fig. 4.

The figure shows the amount of connections that must be created for a successful attack on the model- in all three types of attacks. For example, for the very strong attack, the size of a spammer network must contain more than 14 million users. In this figure we also see the size of the spammer network after t_{spm} from the time the attack network was created, when the attack can actually take place. Since there is an annually growth of approximately 10% users per year in OSN (specifically in Facebook) [17], the number of edges in the spammer network grows. In Fig. 4. E^ψ after T^{AUA} demonstrate this growth after two years. Accordingly, the T^{TF} grows along time, dynamically, forcing the spammer network to add more users, thus making the attack harder, and non-realistic in OSN terms. For an optimized attack, Fig. 5. demonstrates the effort required to attack

Table 1. Experimental results for trust values for the model's parameters.

Parameter	Attribute	Experimental value
T^{AUA}	Age of User Account (OSN seniority)	23.82
T^{TF}	Total Friends	244.34
T^{MF}	Mutual Friends	37
T^{FD}	Friendship Duration	17.12
MTV	Minimal Trust Value	0.745

**Fig. 4.** Spammer network sizes of different attacks**Fig. 5.** Optimized attack complexity

networks with connectivity level of 0.5. We can see that the number of steps required to find the minimal vertex cover is very high relative to the size of the network being attacked. The implementation of the model, with these relevant threshold values for the parameters is meant to be performed by the OSN administration, per each user's network.

6 Discussion, Conclusion, and Future Work

The problem of attacks by malicious users in OSN has many aspects and applications. Using several aspects in a comprehensive Trust-based model that was presented in this paper is a genuine necessity for OSN privacy. In this research we have established the strength of the comprehensive model by analyzing the possible attack scenarios of creating a spammer community that may "contaminate" the model's raw attributes. These attributes are hard to fake since they are built on real OSN user presence and real numerical assets. The comprehensive coverage of Access Control, Flow Control and Trust provides a solid infrastructure for OSN privacy. We have simulated several attack scenarios based on the preliminary evaluations of the properties from our previous research and show that the effort required by the attacker make these attacks infeasible. In current and future work, we are refining the model by considering both the categories and context of data instances and learning User profile from past actions in different contexts.

References

1. Voloch, N., Levy, P., Elmakies, M., Gudes, E.: A role and trust access control model for preserving privacy and image anonymization in social networks. In: IFIPTM 2019 - 13th IFIP WG 11.11 International Conference on Trust Management (2019)
2. Voloch, N., Levy, P., Elmakies, M., Gudes, E.: An access control model for data security in online social networks based on role and user credibility. In: International Symposium on Cyber Security Cryptography and Machine Learning (CSCML 2019). Springer, Cham (2019)
3. Gudes, E., Voloch, N.: An information-flow control model for online social networks based on user-attribute credibility and connection-strength factors. In: CSCML 2018, 2nd International Symposium on Cyber Security Cryptography and Machine Learning (2018)
4. Voloch, N., Gudes, E.: An MST-based information flow model for security in online social networks. In: The 11th IEEE International Conference on Ubiquitous and Future Networks (ICUFN 2019) (2019)
5. Heatherly, R., Kantarcioglu, M., Thuraisingham, B.: Preventing private information inference attacks on social networks. *IEEE Trans. Knowl. Data Eng.* **25**(8), 1849–1862 (2012)
6. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. *Computer* **29**(2), 38–47 (1996)
7. Lavi, T., Gudes, E.: Trust-based Dynamic RBAC. In: Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP), pp. 317–324 (2016)
8. Li, Z., Shen, H., Sapra, K.: Leveraging social networks to combat collusion in reputation systems for peer-to-peer networks. *IEEE Trans. Comput.* **62**(9), 1745–1759 (2012)
9. Sun, J., Zhu, X., Fang, Y.: A privacy-preserving scheme for online social networks with efficient revocation. In: 2010 Proceedings IEEE INFOCOM, pp. 1–9. IEEE (2010)
10. Viswanath, B., Bashir, M.A., Crovella, M., Guha, S., Gummadi, K.P., Krishnamurthy, B., Mislove, A.: Towards detecting anomalous user behavior in online social networks. In: 23rd {USENIX} Security Symposium ({USENIX} Security 2014), pp. 223–238 (2014)
11. Sirur, S., Muller, T.: The reputation lag attack. In: IFIP International Conference on Trust Management, pp. 39–56. Springer, Cham (2019)
12. Lee, K., Caverlee, J., Webb, S.: The social honeypot project: protecting online communities from spammers. In: Proceedings of the 19th International Conference on World Wide Web, pp. 1139–1140 (2010)
13. Paradise, A., Shabtai, A., Puzis, R., Elyashar, A., Elovici, Y., Roshandel, M., Peylo, C.: Creation and management of social network honeypots for detecting targeted cyber attacks. *IEEE Trans. Comput. Soc. Syst.* **4**(3), 65–79 (2017)
14. Huber, M., Mulazzani, M., Weippl, E., Kitzler, G., Goluch, S.: Friend-in-the-middle attacks: exploiting social networking sites for spam. *IEEE Internet Comput.* **15**(3), 28–34 (2011)
15. Shrivastava, N., Majumder, A., Rastogi, R.: Mining (social) network graphs to detect random link attacks. In: 2008 IEEE 24th International Conference on Data Engineering, pp. 486–495. IEEE (2008)
16. Dinur, I., Safra, S.: On the hardness of approximating minimum vertex cover. *Ann. Math.* 439–485 (2005)
17. <https://www.businessofapps.com/data/facebook-statistics/>



Media Partisanship During Election: Indonesian Cases

Ardian Maulana^(✉)  and Hokky Situngkir 

Department of Computational Sociology, Bandung Fe Institute, Bandung 40151, Indonesia
ai@compsoc.bandungfe.net

Abstract. Analysis of media partisanship during election requires an objective measurement of political bias that frames the content of information conveyed to the audience. In this study, we propose a method for political stance detection of online news outlets based on the behavior of their audience in social media. The method consists of 3 processing stages, namely hashtag-based user labeling, network-based user labeling, and media classification. Evaluation results show that the proposed method is very effective in detecting the political affiliation of Twitter users as well as predicting the political stance of news media. Overall, the stance of media in the spectrum of political valence confirms the general allegations of media partisanship during the 2019 Indonesian election. Further elaboration regarding news consumption behavior shows that low-credibility news outlets tend to have extreme political positions, while partisan readers tend not to question the credibility of the news sources they share.

Keywords: Media network · Media partisanship · Twitter · Label propagation algorithm

1 Introduction

The rapid development of online media and social media in recent years has radically changed the way people consume information. Survey shows that 63% of people read news digitally [1], while social networks, such as Twitter and Facebook, are the platforms where people share and discuss the latest news. The combination of online news media and social one strengthens the role of news outlets as gatekeepers of information concerning the formation of public opinion [2, 3].

The neutrality of news media is difficult to maintain at the time of the election. This has increasingly become a public concern that given the ability of news media to influence individual choices, which possibly become an impact on the outcome of the election. The scientific efforts to examine the partisans' behavior of news outlets during the election are constrained by the lack of data about the ideological stance of news media [4–7]. The majority of news outlets do not openly express their political positions on various issues [5]. Generally, media alignment is reflected through content, terminology, and arguments used in framing reported issues. In consequence, it is difficult to objectively measure the political biases contained in the media frame. The alternative approach is to infer

the stance of media based on the political affiliation of their audiences. This approach is based on the assumption that people naturally tend to interact with information adhering to their preferred narrative [6–8].

Social networks like Twitter are very rich in information related to user behavior, e.g. tweet contents, followers, hashtags used. This information can be used to identify users' political affiliations, as well as the political leaning of news outlets they read. In this study, we propose a method for political stance detection of online news outlets based on the behavior of their audience in social media. The method consists of 3 processing stages, as follow: (i) Hashtag-based user labeling: we use a number of political hashtags, i.e. hashtags that are strongly associated with certain political groups, to infer political affiliations of users of these hashtags; (ii) Network-based user labeling: we expand the number of tagged users using Label Propagation Algorithm; (iii) Media classification: at this stage, we use polarity rule to identify the political stance of news outlets based on the political affiliation of their audiences.

We applied this methodology to the tweet dataset related to the 2019 Indonesian general election, to observed media alignments during the election. In doing so, we also report news consumption patterns on Twitter concerning credibility and partisan behavior of news sources. This paper is structured as follows: sections two and three discuss data and analysis methods, results of the analysis will be shown in Sect. 4, while the final section will discuss a number of conclusions and contributions of this study.

2 Data

We conducted the data¹ collection process from 27 March to 19 May 2019, which covered the campaign period, general elections (April 17, 2019), vote recapitulation, and the announcement period (May 21, 2019). Table 1 shows the descriptive statistics of the data used in this study. Tweet data was extracted from Twitter using the DMI-Tcat application [9] based on a number of keywords related to the candidates, namely: (i) Candidate I (Prabowo-Sandiaga Uno): prabowo, sandiaga uno; (ii) Candidate II (Joko Widodo-KH. Maaruf Amin): joko widodo, jokowi, ma'ruf amin, kiai ma'ruf.

Table 1. Descriptive statistics of the dataset

Statistics	Count
# of tweets	13990975
# of tweets with a URL	667821
# of hashtags	74515
# of unique users	3958817

¹ The dataset used in this study is available in limited form at <https://github.com/ardianeff/indome-dialection2019>

3 Method for Political Stance Detection of the Online News Outlets

The process of political stance detection consists of 3 stages: (i) hashtag-based user labeling; (ii) network-based user labeling); and (iii) media classification.

3.1 Hashtag-Based User Labeling

In order to analyze the political stance of news outlets we first find the stances of Twitter users. Twitter users usually use political hashtags in their tweets to express their support for the political message contained in the hashtag. Nowadays, political hashtags are kind of strategies commonly used to mobilize opinions, popularize the candidates, or attack the opponents [10]. In this study, we use this simple fact to identify the political affiliations of Twitter users. Figure 1 shows a histogram of the 10 most widely used political hashtags in the 2019 Indonesian elections.

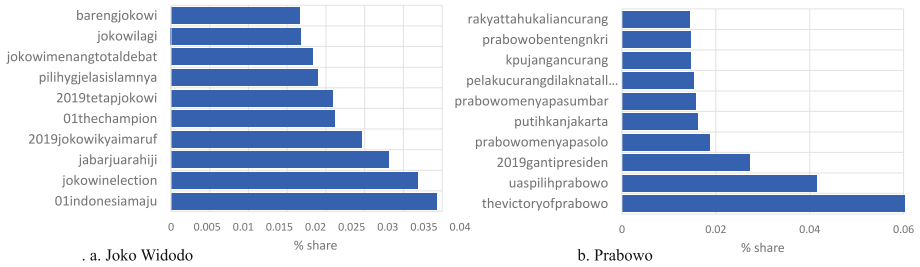


Fig. 1. 10 most used political hashtags used by the candidate: (a) Joko Widodo; (b) Prabowo

We label manually 1400 hashtags, which is 700 for each candidate, of the total 74515 hashtags recorded in the dataset. Each of these hashtags have been used by at least 10 different Twitter users. We apply polarity rule to infer the political affiliation of the users, as follow [11]:

$$V(u) = 2 \frac{\frac{tf(u, C_0)}{total(C_0)}}{\frac{tf(u, C_0)}{total(C_0)} + \frac{tf(u, C_1)}{total(C_1)}} - 1 \tag{1}$$

where $tf(u, C_0)$ is the number of times (user frequency) user u use group of the hashtag C_0 of candidate i , $total(C_0)$ is a sum of the frequency of hashtag usage by all users. The hashtag of other candidates is defined similarly. The political valence value $V(u)$ is in the range $-1 \leq V(u) \leq 1$. To ensure the user’s political affiliation, we use a threshold value of ± 0.2 , where users are lean-to Prabowo if they have a valence score < -0.2 , while lean-to Joko Widodo if the valence score > 0.2 . Table 2 shows that at this stage we are able to identify the political affiliation of 181,145 Twitter users, of which 89,000 are Jokowi’s supporters and 92,145 are Prabowo’s supporters.

Table 2. Identification of users’ political affiliations using hashtags and network-based labeling

Label	Hashtag	Network
pro-Joko Widodo	89000	176109
pro-Prabowo	92145	366134
Total	181145	542243

3.2 Network-Based User Labeling

A central assumption in this stage is that if a user retweets a tweet, they most likely agree or endorsed message contained in that tweet. Some empirical studies [7, 12, 13] showed that content consumption in social media is dominated by selective exposure (i.e., the tendency of users to ignore dissenting information and to interact with information adhering to their preferred narrative). It means that individuals tend to selectively interact, which is only with other individuals who share their political understanding. In this stage, we first construct an undirected weighted retweet graph to represent an interaction between Twitter users, where vertices represent users and directed relationships between vertices are formed if one user retweets another user’s posts. Table 3 shows descriptive statistics of this network, where the density value indicates that this network is a sparse network, where largest component consisting of 542,243 nodes.

Table 3. Descriptive statistics of the retweet network

Statistics	Retweet network	Giant component
# of nodes	558801	542243
# of edges	4372893	4372706
Density	2.8E−05	2.97E−05
Average degree	15.651	16.1282
# of component	16397	1

Then, we apply the label propagation algorithm to classify each node in the network as pro-Joko Widodo or pro-Prabowo. In this study we do not consider the existence of non-polarized users by assuming that each user is exposed to political information and therefore will have a tendency towards one of the candidates [11, 14]. Furthermore, supporters of both candidates who are less polarized tend to consume media that is considered politically neutral, and hence will place these media in the middle of the political spectrum. We accommodate this latter possibility by establishing a ‘moderate’ media type in our media classification (see Eq. 2).

Label propagation algorithms are graph-based semi-supervised learning methods, and use the label information of labeled data to predict the label information of unlabeled data. At this stage, we used 153,990 labeled nodes identified in the previous stage as seeds (the list of labeled nodes). We fix the seeds’ labels so they do not change in the process,

since this seed list also serves as our ground truth. This algorithm works iteratively to renew the label of each node based on the majority label of its neighbor node. This process is carried out until the labels of the majority of nodes no longer change [14, 15].

The k-stratified cross (5-fold) validation model is implemented to the set of 153,990 seeds to validate result of the label propagation algorithm [14]. We use 4/5 of the seed nodes as training data, while the remaining nodes are used to evaluate the algorithm performance. The evaluation results in Table 4 show prediction accuracy of ~98%. This strengthens confidence in the performance of the classification algorithm that we use. At this stage, we successfully identified the political affiliation of 388,253 users in the retweet network.

Table 4. Mean (standard dev.) score of label propagation algorithm performance.

Precision	Recall	Accuracy
0.98787 (0.005)	0.983108 (0.006)	0.984955 (0.006)

3.3 Media Classification

The political affiliation of Twitter users obtained in the previous stage is used to predict the political stance of news media during the election. We determine the stance of a media based on the average political affiliation of Twitter users who are those media audiences (see Eq. 1) [11]. In other words, the political alignments of a news outlet are measured by the extent to which these outlets become sources of information for one political group. The greater the audience share of an outlet has come from a particular political group, the stronger the association between the two. As such, the score of media alignments capture differences in the type of content, which covers topics and news frames, shared by partisan users.

We split the alignment scores into 5 equal size bands, as follows [8]:

$$S(v) = \begin{cases} -2 & \text{if } v \in [-1, -0.6] \\ -1 & \text{if } v \in [-0.6, -0.2] \\ 0 & \text{if } v \in [-0.2, 0.2] \\ 1 & \text{if } v \in [0.2, 0.6] \\ 2 & \text{if } v \in [0.6, 1] \end{cases} \quad (2)$$

where $S(v) < 0$ means that the media tends to lean to Prabowo, $S(v) > 0$ means the media tends to lean Joko Widodo, and $S(v) = 0$ means that the media tends to be politically neutral or moderate news media.

4 Analysis

Figure 2 shows the daily number of articles and unique articles shared by users during the data collection period. The statistics of the unique article become a proxy for the volume

of articles published by the media outlets. In general, the two indicators do not show different dynamics. This indicates the influence of the media on the intensity of news consumption on social media. Although the daily volume has fluctuated, the dynamics clearly show an upward trend ahead of the election. This indicates election-related news, as well as the reader’s attention, is increasing toward the election, which reaches its peak on election day, then shows a downward trend afterward.

In this study, we only investigated 560 news outlets out of 700 outlets found in the dataset. Overall, we only focus on domestic news media, which has been shared by 10 different Twitter users.

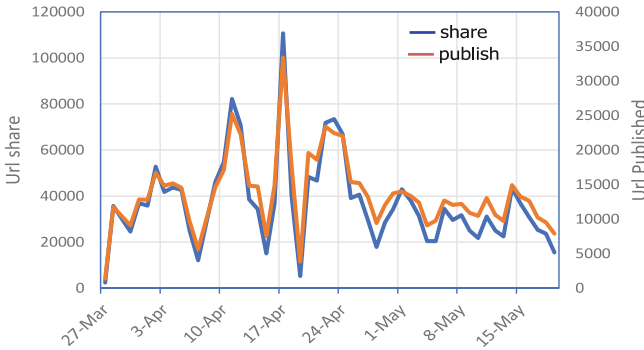


Fig. 2. Daily number of articles and unique articles shared by twitter users. Dotted lines are trend lines.

4.1 The Political Stance of News Media Outlets

Figure 3 shows the distribution of media stance in the 2019 Indonesian elections. The bimodal pattern indicates the presence of media polarization where the majority of news outlets reside on the extreme side of the political spectrum. From this figure, it is known that the proportion of Joko Widodo-leaning media is greater than the number of news

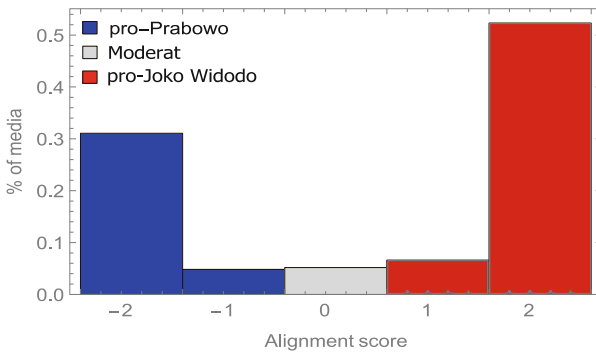


Fig. 3. Distribution of political alignment scores

media in favor of Prabowo. However, Prabowo-leaning media is superior in terms of frequency of share and total users.

Figure 4 shows the position of a number of mainstream media in the spectrum of political alignments. From this figure, it is known that political valence scores are able to capture the main differences between news outlets on each side of the spectrum, as follow:

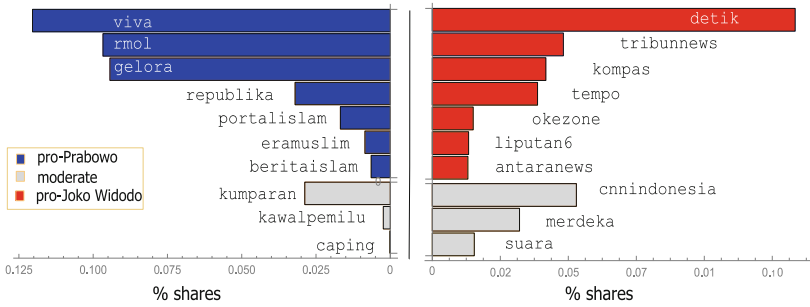


Fig. 4. The position of a number of mainstream media in the spectrum of political alignments: (left) pro-Prabowo; (right) pro-Joko Widodo.

- Majority of Islamic news media, such as Republika, Portal-Islam, Era Muslim, Konten Islam tend to favor Prabowo. This is not surprising because religious issues are very dominant in the 2019 Indonesian elections, and Prabowo was imaged as a representative of an Islamic group;
- The opposition news outlets which has criticized the Joko Widodo government for the past 5 years, such as Viva, Rmol, Gelora has a valence score on Prabowo;
- Most of the mainstream news media, such as Kompas, Detik and Tempo tend to support Joko Widodo. While some others such as CNN, Merdeka, tend to be politically moderate.

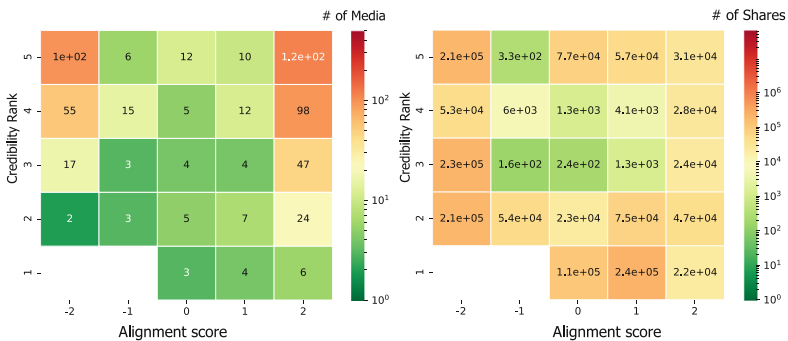


Fig. 5. Heat map between political valence score vs credibility ranking by: (left) number of media; (right) number of shares.

The heat map shown in Fig. 5 illustrates the relationship between the political alignments of news media and their credibility. In this study we use Alexa Rank [16] as a proxy to assess the credibility of a media.

As shown in Fig. 5(left), most of mainstream media with high credibility ratings have a neutral valence score or tend to favor Joko Widodo, while Prabowo-leaning media generally have moderate or low credibility. In addition, most of low-credibility media tend to have extreme political valence scores. In other words, low-credibility media tend to be more partisan than the one with high credibility. From Fig. 5 (right) we also know that, for all political valence scores, the intensity of information dissemination originating from low-credibility media is relatively not much different compared to high-credibility media. This means that partisan readers tend not to question the credibility of the news sources they share. We highlight this empirical fact related to the rise of false news during the election and the potential of low-credibility media as sources of misinformation.

5 Conclusion

In this study, we use the partisan behavior of media audiences on Twitter to identify political alignments of news media during the 2019 Indonesian elections. The identification method we proposed is carried out in 3 stages, as follow: (i) Identification of the political affiliations of twitter users based on the political hashtag they used in their tweet; (ii) Identification of the political affiliations of Twitter users based on their interaction networks using the label propagation algorithm; (iii) Identification of the political alignments of news media based on the political affiliation of its audience. Evaluation results show that the proposed method is very effective in detecting the political affiliation of Twitter users as well as predicting the political stance of news media. The position of media in the spectrum of political valence confirms the general allegations of media partisanship during the election. Further elaboration regarding news consumption behavior shows that low-credibility news outlets tend to have extreme political positions, while partisan readers tend not to question the credibility of the news sources they share.

References

1. Nic, N., Levy, D.A.L., Nielsen, R.K.: Reuters Institute Digital News Report 2018. Reuters Institute Digital News (2018)
2. Vos, T.P.: Journalists as Gatekeepers. In: Wahl-Jorgensen, K., Hanitzsch, T. (eds.) *The Handbook of Journalism Studies*, pp. 90–104. Routledge (2019)
3. Allcott, H., Gentzkow, M.: Sosial media and fake news in the 2016 election. *J. Econ. Perspect.* **31**, 211–236 (2017)
4. Groeling, T.: Media bias by the numbers: challenges and opportunities in the empirical study of Partisan news. *Ann. Rev. Polit. Sci.* **16**(1), 129–151 (2013)
5. Barberá, P., Sood, G.: Follow your ideology: measuring media ideology on social networks. In: *Annual Meeting of the European Political Science Association*, Viena (2015)
6. Becatti, C., Caldarelli, G., Lambiotte, R., Saracco, F.: Extracting significant signal of news consumption from social networks: the case of twitter in italian political elections. *Palgrave Commun.* **5**(1) (2019)

7. Bakshy, E., Messing, S., Adamic, L.A.: Exposure to ideologically diverse news and opinion on facebook. *Science* **348**(6239), 1130–1132 (2015)
8. Stefanov, P., Darwish, K., Atanasov, A., Nakov, P.: Predicting the topical stance and political leaning of media using tweets. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2020)
9. Borra, E., Rieder, B.: Programmed method: developing a toolset for capturing and analyzing tweets. *Aslib J. Inf. Manag.* **66**(3), 262–278 (2014)
10. Varol, O., Uluturk, I.: Journalists on twitter: self-branding, audiences, and involvement of bots. *J. Comput. Soc. Sci.* **3**(1), 83–101 (2020)
11. Conover, M., Ratkiewicz, J., Francisco, M., Gonçalves, F., Menczer, F., Flamini, A.: Political polarization on twitter. In: Proceedings of 5th International Conference on Weblogs and Social Media (2011)
12. Vicario, M.D., Bessi, A., Zollo, F., Petroni, F., Scala, A., Caldarelli, G., Stanley, H.E., Quattrociocchi, W.: The spreading of misinformation online. *Proc. Natl. Acad. Sci. U.S.A.* **113**(3), 554–559 (2016)
13. Zollo, F., Bessi, A., Del Vicario, M., Scala, A., Caldarelli, G., Shekhtman, L., Havlin, S., Quattrociocchi, W.: Debunking in a world of tribes. *PLoS ONE* **12**(7), e0181821 (2017)
14. Badawy, A., Ferrara, E., Lerman, K.: Analyzing the digital traces of political manipulation: the 2016 Russian interference twitter campaign. In: Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2018) (2018)
15. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* **76**(3) (2007)
16. Alexa Internet: Keyword Research, Competitor Analysis RESEARCH & Website Ranking. <https://www.alexa.com>. Accessed 10 Aug 2019



Media Polarization on Twitter During 2019 Indonesian Election

Ardian Maulana^(✉)  and Hokky Situngkir 

Department of Computational Sociology, Bandung Fe Institute, Bandung 40151, Indonesia
ai@compsoc.bandungfe.net

Abstract. In this study, we investigate the phenomenon of political polarization on news consumption patterns of Twitter users during 2019 Indonesian elections. By modeling news consumption as a bipartite network of news outlets-Twitter users, we observed the emergence of a number of media communities based on audience similarity. By measuring political alignments of each news outlet, we reveal the politically fragmented landscape of Indonesian news media, where each media community becomes an political echo-chamber for its audience. Our findings highlight the important role of mainstream media as a bridge of information between political echo-chamber in social media environment.

Keywords: Media network · Echo-chamber · Community detection · Twitter · Election

1 Introduction

The outbreak of extreme political polarization in various democratic countries has been the problem of this century [1]. This phenomenon cannot be separated from the rise of digital information space, e.g. social media, online news media, which makes it easier for citizens to access and discuss political information [2]. On the one hand, The combination of online news media and social one increases the chances of individuals being exposed to information from a variety of perspectives [3]. But on the other hand, mediation and personalization of information by social networks also has the potential to limit exposure to only information that is politically agreed upon [4], giving rise to misperceptions of facts and events [5] and leading to the emergence of increasingly extreme political attitudes [6].

A number of studies have shown empirically the tendency of social media users to focus on specific narratives, and interact intensively with those who have the same political preferences [7–9]. This micro tendency may lead to the emergence of echo-chambers [7, 10] that divide the social media space into politically homogeneous user communities [11]. In each community, users tend to ignore dissenting information and to interact with information adhering to their preferred narrative. The study of digital echo-chamber phenomena is quite challenging [7–12]. However, most of research in this area examine fragmentation and polarization in user networks. Meanwhile, empirical works to investigate information segregation due to fragmentation of information sources is constrained by the difficulty of measuring the political tendencies of news media [13].

In network perspective, the dynamics of information consumption on social media is basically a process of network formation that connects social media users and information sources (e.g. web, blogs, online media, etc.) through various means, e.g. browsing, sharing and others. Therefore, in this study, to gain a better understanding about information echo-chamber and its polarization effect during election, we will explore the anatomy of Indonesian media network constructed from news consumption activity on Twitter during 2019 Indonesian General Election. Specifically, we analyzed 667,821 election-related tweets to investigate the phenomenon of polarization of the news media in Indonesia, as well as explore the role of each news outlet in the dynamics of news consumption during the election. This paper is written with the following structure: data and analysis methods will be discussed in sections two and three of this paper, while in section four we discuss the results of the analysis based on the objectives of this study. Conclusions and contributions of this study are discussed at the end of this paper.

2 Data

This study investigates news consumption patterns on Twitter during the 2019 Indonesian presidential and legislative elections. We conducted the data collection process from March 27 to May 19, 2019, covering the campaign period, elections (April 17, 2019), the vote recapitulation and announcement period (May 21, 2019). We use DMI-Tcat application [14] to extract data from Twitter based on a number of keywords related to the candidates, namely: (i) Candidate I (Prabowo - Sandiaga Uno): prabowo, sandiaga uno; (ii) Candidate II (Joko Widodo - KH. Maaruf Amin): joko widodo, jokowi, maaruf amin, kiyai maaruf.

Table 1 shows the descriptive statistics of the tweet dataset¹ used in this study. Overall, we only focused on 667,821 of total 13,990,975 tweets, which contained news links from 559 Indonesian news media, and were shared at least ten times by Twitter users.

Table 1. Descriptive statistics of the 2019 Indonesian Election tweet dataset

Statistics	Count
# of tweets	13,990,975
# of tweets with a Url	667,821
# of hashtags	74,515
# of unique users	3,958,817

¹ The dataset used in this study is available in limited form at <https://github.com/ardianeff/indomediaelection2019>.

3 Method

3.1 Bipartite Network

News consumption activity on Twitter can be modeled as a bipartite network between Twitter users and information sources. The user-media bipartite network (S) is composed of two type of nodes, namely user node ($n_A = 115,621$) and news outlet node ($n_B = 559$). Each edge ($n_e = 466,542$) connecting those two nodes indicates that a user a_i ($a_i \in A$) consumes news, which is expressed through sharing or retweeting, from outlet b_i ($b_i \in B$).. To explore connectivity patterns among news outlets, we project bipartite network S into news media network \hat{S} , where edge weight indicates a number of shared audience between two outlets. In this study, we focus on the largest connected component of weighted network \hat{S} , which is composed of 559 media nodes and 55,662 edges.

Table 2 shows that the projection network \hat{S} has a fairly dense structure ($\rho = 0.35$). Therefore we need to evaluate significance of each edge and filter out random interaction between twitter users and news sources. In this study we use the method proposed by [15], which has been proven effective for investigating bipartite systems in various areas. Specifically, we attach p-values at each edge of the projection network, then apply multiple hypothesis testing using a statistical threshold value of 0.01, and then make moderately corrections using False Discovery Rate method (FDR) [16].

Table 2. Characteristics of Indonesian news media network

Statistics	Pre-filtered network	Final network
# of node	559	528
# of edge	55,662	27,192
Diameter	3	6
Ave. path length	1.64	1.953

3.2 Community Structure

In this study we use Fast Greedy algorithm [17] to analyze the meso structure of projection network \hat{S} . As shown in Table 3, this algorithm revealed five media communities, where there were two very dominant clusters, covering 98% of total news outlets analyzed. To validate the results, we also implemented the Walk Trap algorithm [18], then compared the results of both algorithm using Adjusted Rand Index (ARI) method [19]. We find the ARI coefficient is 0.8, which indicates that two algorithms produces similar result.

Table 3. Descriptive statistics of community partitions using the Fast Greedy algorithm

Statistics	Fast Greedy
# of community	5
Modularity	0.256453
# node in cluster 1	202
#node in cluster 2	313
# node in other clusters	13

3.3 Political Stance of Online News Media

We need to measure political stance of news outlets in order to investigate political polarization that occurs in media networks during the elections. In this study the media stance was identified based on partisan behavior of their audience, assuming that people tended to be selective about information, i.e. only reading and sharing news articles in accordance with their political preferences. Following [20], the process of media classification is carried out in 3 stages, as follow: (i) Hashtag-based user labeling: 1400 political hashtag associated with certain political groups are used to identify the political affiliations of these hashtag users. At this stage, we successfully identified 153,990 labeled users, which will then be used as seed nodes for the label propagation algorithm at later stage; (ii) Network-based user labeling: at this stage we apply Label Propagation algorithm to expand the number of labeled users [21, 22]. We have successfully identified political affiliation of 388,253 Twitter users, with prediction accuracy of ~ 0.98 ; (iii) Media classification: we use polarity rule [11] to identify media stance based on the political affiliation of their audiences. Table 4 shows classification result of 560 Indonesian news outlets.

Table 4. Composition of partisan media within each media community

	# of media	# of share	# of user
pro-Joko Widodo	330	373932	39806
Neutral	29	228522	87074
pro-Prabowo	201	404058	61966

4 Analysis

4.1 News Consumption Pattern

The current disintermediated environment composed by a heterogeneous mass of information sources, on the one hand, has reduced the centralization of information, which

is a characteristic of information consumption patterns in the previous era [23]. But on the other hand, it may lead to the emergence of audience fragmentation into various groups of information sources [24, 25]. The distribution of readers in Fig. 1 shows that people tend to interact with a few news outlets, despite the availability of various alternative news sources. Naturally, this may lead to a wider-but-fragmented landscape of information sources, where news outlets are grouped based on audience similarity.

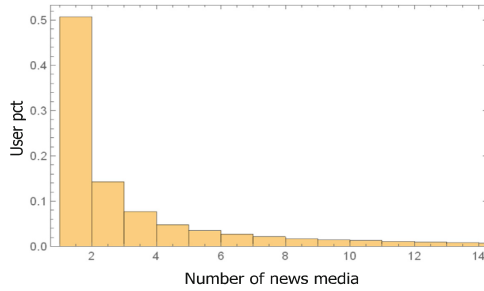


Fig. 1. The number of news media consumed by Twitter users

4.2 Segregation in Media Network

The Fast Greedy algorithm has successfully identified two dominant communities in the Indonesian news media network, covering 98% of the total news outlets analyzed. This media network has a high value of modularity ($M = 0.25$), which indicates a segregation of information sources in the news media landscape during the election. Considering that the grouping of news outlets emerge from the interaction between audience and news sources, it is necessary to measure the extent to which segregation occurs between the two dominant media communities, as follow [26, 27]:

$$p(u) = \frac{y - x}{y + x} \tag{1}$$

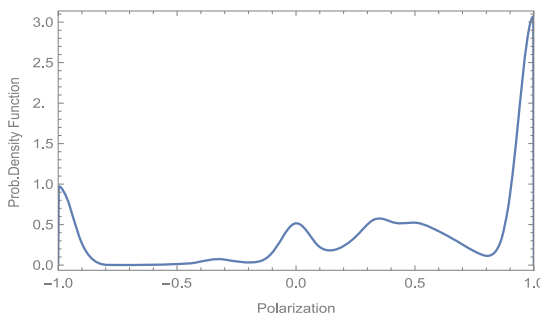


Fig. 2. Audience segregation

where $y(x)$ is a fraction of twitter users who share news tweets from outlets in media community $C_1(C_2)$. Figure 2 shows the presence of strong bimodality in the distribution of news audience activity in each community. This indicates that each media community is an echo-chamber for their respective audiences, that is a groups of like-minded people cooperating to reinforce a shared narrative.

4.3 Political Polarization

To understand the relationship between segregation in news media networks and the political alignments of news outlets during the election, we elaborate the composition of partisan media in the two dominant media clusters. As shown in Table 5, the composition of the partisan media in each community tends to be politically homogeneous. This fact confirms the occurrence of political polarization in Indonesian media networks. Table 5 also shows that Joko Widodo-leaning media has a stronger tendency to group in the same cluster than Prabowo-leaning media, while news outlets with moderate political stance are relatively small in number and spread evenly in two dominant clusters.

Table 5. Composition of partisan media within each media community.

Political alignment	Cluster I	Cluster II
pro-Prabowo	0.837	0.045
Moderate	0.089	0.028
pro-Joko Widodo	0.074	0.927

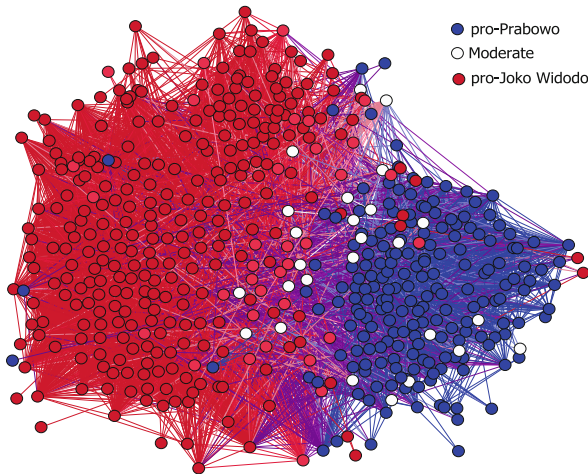


Fig. 3. Political polarization in Indonesian news media during 2019 General Elections

Figure 3 visualizes the political polarization of Indonesian news media during 2019 Presidential Elections. As seen in the figure, the structure of media network is divided into two dominant clusters, each with a relatively homogeneous political identity. This shows that the pattern of news consumption in 2019 Indonesian elections is not only fragmented, but also forms a political echo-chamber where audiences tend to be exposed to politically homogeneous information coming from news outlet with the same political tendencies.

4.4 Interaction Across Political Communities

We then elaborate on empirical facts about interactions between news media across political affiliations [20–22, 26–28]. Figure 4 shows the statistical characteristics of interaction between news outlets, intra and inter media communities. In general, the Indonesian news media network have homophily properties, where news outlets with the same political stance tend to be connected to each other (Joko Widodo-leaning media: median = 0.842, IQR = [0.809, 0.873]; Prabowo-leaning media: median = 0.543, IQR = [0.514, 0.577]). In general, this characteristic is stronger for Joko Widodo-leaning media than Prabowo-leaning media. Furthermore, the interaction tendency from Joko Widodo-leaning media to Prabowo-leaning media (median = 0.045, interquartile distance (IQR) = [0.0023, 0.064]) is much smaller than the opposite (median = 0.306, IQR = [0.273, 0.323]). Meanwhile, interactions tendency from moderate news media partisan media are

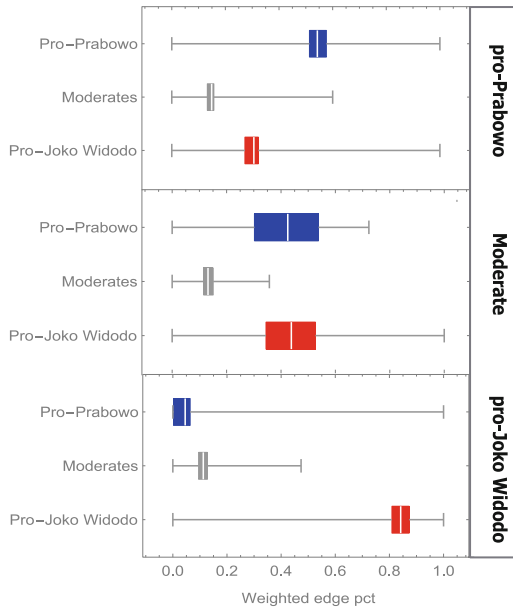


Fig. 4. Statistical characteristics of interaction between news outlets, intra and inter political affiliations. White vertical lines are median values, horizontal thick lines are interquartile ranges, and black thin horizontal lines are 10th and 90th percentile values.

almost equal (pro Joko Widodo: median = 0.425, IQR = [0.345, 0.527]; pro Prabowo: median = 0.438, IQR = [0.302, 0.538]).

The statistics of interaction between media across political affiliations indicate that information exposure to Joko Widodo's supporters is relatively more homogeneous, coming from news outlets with the same political affiliation, compared to information exposure to Prabowo's supporters. This can be understood by looking at the composition of partisan media in each media community (see Table 5). Moreover, as discussed in previous studies [20], Prabowo-leaning media is dominated by segmented news media, such as the Islamic news outlets (e.g. *eramuslim*, *portal-islam*, *republika*) and opposition media (e.g. *viva*, *rmol*, *gelora*) while mainstream media tends to be neutral or in favor of Joko Widodo. As a result, Prabowo's supporters tend to be exposed to information coming from the pro-Joko Widodo news media, but not vice versa.

4.5 News Media Centrality

We further investigate the role and position of each news outlet in the information ecosystem during the 2019 Indonesian elections. We use two indicators, namely within module degree z-score (z_i) and participation coefficient (Pc_i) [29], to measure the role of a news outlet based on its relations with other outlets within or between media communities. The within module degree z-score (z_i) measures connectivity of a news outlet in its community internally, as follow:

$$z_i = \frac{k_i - \bar{k}_{s_i}}{\sigma_{k_{s_i}}} \quad (2)$$

where k_i is the degree of news outlet i within the cluster s_i , \bar{k}_{s_i} is the average degree of all media in cluster s_i , and $\sigma_{k_{s_i}}$ is the standard deviation of degree k in cluster s_i . The greater the value of z_i , the higher the connectivity of outlet i relative to other outlet in its community. Meanwhile, cross-cluster node connectivity is evaluated using the participation coefficient (pc_i) indicator, as follows:

$$Pc_i = 1 - \sum_{s=1}^M \left(\frac{k_{is}}{k_i} \right)^2 \quad (3)$$

where k_{is} is the number of relation of outlet i to other outlets in cluster s . Value $Pc_i \sim 0$ if outlet i is only connected to the outlet in its group only. Conversely, the value of $Pc_i \sim 1$ if the relation of an outlet is evenly distributed in all clusters. The combination of those two indicators forms 7 regions of node roles within z-P parameter space, namely (i) R1: ultra-peripheral nodes ($z < 2.5, P \leq 0.05$); (ii) R2: peripheral nodes ($z < 2.5, 0.05 \leq P \leq 0.62$); (iii) R3: non-hub connector ($z < 2.5, 0.62 \leq P \leq 0.8$); (iv) R4: non-hub kinless nodes ($z < 2.5, 0.8 \leq P$); (v) R5: provincial hubs ($z \geq 2.5, P \leq 0.3$); (vi) R6: connector hubs ($z \geq 2.5, 0.3 < P \leq 0.75$); (vii) R8: kinless hubs: ($z \geq 2.5, P > 0.75$).

As shown in Table 6, it is known that ~96% of news outlets are ultra-peripheral (R1) or peripheral nodes (R2), or low degrees nodes with little or no cross-cluster connection. The remaining media fills the R5 region as a provincial hub and the R6 region as connector hubs.

Table 6. Descriptive statistics of the news media role during election. Media rating is based on Alexa rank [30]. (JW: Joko Widodo; P: Prabowo; M: Moderate).

R	News outlets	# of outlets	Composition (%)	Median of media rating
1	kanlagi, indosport, time, apnews, voanews, thejakartapost, foreignpolicy, cgtn, paperform, historia, etc.	56	JW: 56.4 M: 1.8 P: 41.8	1,050,000
2	grid, suara, brilio, kompasiana, bolasport, cnbcindonesia, wowkeren, dream, bola, abc, etc.	449	JW: 58.4 M: 5.12 P: 36.5	687,211
5	gelora, rmol	2	JW: 0 M: 0 P: 100	1,070,000
6	Okezone, tribunews, detik, kompas, liputan6, sindonews, kumparan, idntimes, merdeka, tempo, etc.	21	JW: 76.2 M: 14.3 P: 9.6	918

How the partisan outlets filled the R5 and R6 regions revealed differences in the information echo space characteristics of the two candidates. As shown in Table 6, there are only two news outlets in the R5 region, and both are Prabowo-leaning media. This means that *gelora* and *rmol* are central outlets within the information echo-chamber of Prabowo's supporter. However, this also implies that information structure of Prabowo's media community is more centralized than Joko Widodo's media community. In general, Joko Widodo-leaning media, as well as moderate media, dominate the R6 region as a connector hubs, which means that these outlets are consumed by supporters of both candidates. As shown in Table 6, news outlets in the R6 region have a high median Alexa Rank, which indicates this region is dominated by mainstream news media. This fact highlights the important role of mainstream news media as a bridge of information between opposite political sides, especially in heated election.

5 Conclusion

In this study we reveal empirical facts about political polarization in Indonesian news media network during 2019 Indonesian General Elections. By modeling news consumption patterns as a bipartite network of news outlets-Twitter users, we observed the emergence of a number of media communities based on audience similarity. By measuring the political alignments of each news outlet, we reveal the politically fragmented Indonesian news media landscape, where each media community becomes an political echo-chamber for its audience. More specifically we find that, compared to the Prabowo media cluster which tends to be centralized in a small number of outlets, Joko Widodo's supporters have diverse sources of information. However, information exposure to Joko Widodo's supporters is relatively more homogeneous coming from the media with the same political affiliations.

Nowadays, the understanding of the impact of social media and online news media on the emergence of extreme polarization in political discourse is one of the most pressing challenges for both science and society. Our finding highlight the important role of mainstream media as a bridge of information between political echo-chamber in social media environment.

References

1. World Economic Forum: Digital Wildfires in a Hyperconnected World. <https://reports.weforum.org/global-risks-2018/digital-wildfires/>. Accessed 21 July 2019
2. Nic, N., Levy, D.A.L., Nielsen, R.K.: Reuters Institute Digital News Report 2018. Reuters Institute Digital News (2018)
3. Bakshy, E., Rosenn, I., Marlow, C., Adamic, L.: The role of social networks in information diffusion. In: Proceedings of the 21st Annual Conference on World Wide Web (IW3C2), pp. 519–528. ACM, Lyon (2012)
4. Pariser, E.: The Filter Bubble: What the Internet Is Hiding from You. Penguin Press, London (2011)
5. Kull, S., Ramsay, C., Lewis, E.: Misperceptions, the Media, and the Iraq War. *Polit. Sci. Q.* **118**(4), 569–598 (2003)
6. Stroud, N.J.: Media use and political predispositions: Revisiting the concept of selective exposure. *Polit. Behav.* **30**(3), 341–366 (2008)
7. Quattrociocchi, W., Scala, A., Sunstein, C.R.: Echo Chambers on Facebook. *SSRN Electron. J.* (2018)
8. Bessi, A., Petroni, F., Del Vicario, M., Zollo, F., Anagnostopoulos, A., Scala, A., Caldarelli, G., Quattrociocchi, W.: Viral misinformation: the role of homophily and polarization. In: WWW 2015 Companion - Proceedings of the 24th International Conference on World Wide Web, pp. 355–356. ACM, Florence (2015)
9. Vicario, M.D., Bessi, A., Zollo, F., Petroni, F., Scala, A., Caldarelli, G., Stanley, H.E., Quattrociocchi, W.: The spreading of misinformation online. *Proc. Natl. Acad. Sci. U. S. A.* **113**(3), 554–559 (2016)
10. Del Vicario, M., Vivaldo, G., Bessi, A., Zollo, F., Scala, A., Caldarelli, G., Quattrociocchi, W.: Echo chambers: emotional contagion and group polarization on Facebook. *Sci. Rep.* **6**(1) (2016)

11. Conover, M., Ratkiewicz, J., Francisco, M., Gonçalves, F., Menczer, F., Flamini, A.: Political polarization on Twitter. In: Proceedings of 5th International Conference on Weblogs and Social Media (2011)
12. Bessi, A., Zollo, F., Del Vicario, M., Scala, A., Caldarelli, G., Quattrociocchi, W.: Trend of narratives in the age of misinformation. *PLoS ONE* **10**(8), e0134641 (2015)
13. Groeling, T.: Media bias by the numbers: challenges and opportunities in the empirical study of Partisan news. *Ann. Rev. Polit. Sci.* **16**(1), 129–151 (2013)
14. Borra, E., Rieder, B.: Programmed method: developing a toolset for capturing and analyzing tweets. *Aslib J. Inf. Manag.* **66**(3), 262–278 (2014)
15. Tumminello, M., Micciché, S., Lillo, F., Piilo, J., Mantegna, R.N.: Statistically validated networks in bipartite complex systems. *PLoS ONE* **6**(3), e17994 (2011)
16. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Stat. Soc.: Ser. B (Methodol.)* **57**(1), 289–300 (1995)
17. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. *Phys. Rev. E* **70**(6) (2004)
18. Pons, P., Latapy, M.: Computing communities in large networks using random walks. *J. Graph Algorithms Appl.* **10**(2), 191–218 (2006)
19. Rand, W.M.: Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **66**(336), 846–850 (1971)
20. Maulana, A., Situngkir, H.: Media partisanship during election: Indonesian cases. MPRA Paper, 101950 (2020)
21. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* **76**(3) (2007)
22. Badawy, A., Ferrara, E., Lerman, K.: Analyzing the digital traces of political manipulation: the 2016 Russian interference Twitter campaign. In: Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2018) (2018)
23. Sunstein, C.R.: #Republic: Divided Democracy in the Age of Social Media. Princeton University Press, Princeton (2017)
24. Webster, J.G., Ksiazek, T.B.: The dynamics of audience fragmentation: public attention in an age of digital media. *J. Commun.* (2012)
25. Gaol, F.L., Matsuo, T., Maulana, A.: Network model for online news media landscape in Twitter. *Information* **10**(9), 277 (2019)
26. Schmidt, A.L., Zollo, F., Vicario, M.D., Bessi, A., Scala, A., Caldarelli, G., Stanley, H.E., Quattrociocchi, W.: Anatomy of news consumption on Facebook. *Proc. Natl. Acad. Sci. U. S. A.* **114**(12), 3035–3039 (2017)
27. Del Vicario, M., Gaito, S., Quattrociocchi, W., Zignani, M., Zollo, F.: News consumption during the Italian referendum: a cross-platform analysis on Facebook and Twitter. In: Proceedings - 2017 International Conference on Data Science and Advanced Analytics (DSAA 2017), Tokyo, pp. 648–657 (2017)
28. Bakshy, E., Messing, S., Adamic, L.A.: Exposure to ideologically diverse news and opinion on Facebook. *Science* **348**(6239), 1130–1132 (2015)
29. Guimerà, R., Amaral, L.A.N.: Functional cartography of complex metabolic networks. *Nature* **433**(7028), 895–900 (2005)
30. Alexa Internet: Keyword Research, Competitor Analysis RESEARCH & Website Ranking. <https://www.alexa.com>. Accessed 10 Aug 2019



Influence of Retweeting on the Behaviors of Social Networking Service Users

Yizhou Yan¹(✉), Fujio Toriumi², and Toshiharu Sugawara¹

¹ Department of Computer Science and Communications Engineering,
Waseda University, Tokyo 169-8555, Japan

y.yan@is1.cs.waseda.ac.jp, sugawara@waseda.jp

² Department of Systems Innovation, The University of Tokyo,
Tokyo 113-8654, Japan

tori@sys.t.u-tokyo.ac.jp

Abstract. Retweeting is a featured mechanism of some social media platforms such as Twitter, Facebook, and Weibo. Users share articles with friends or followers by reposting a tweet. However, the ways in which retweeting affects the dominant behaviors of users is still unclear. Therefore, we investigate the influence of retweeting on the behaviors of social media users from a networked, game theoretic perspective; in other words, we attempt to clarify the ways in which the presence of a retweeting mechanism in social media promotes or diminishes the willingness of users toward posting articles and commenting. We propose a retweet reward game model that has been derived by adding a retweeting mechanism to a reward game, which is a simple social networking service model. Subsequently, we conduct some simulation-based experiments to understand the effects of retweeting on the behaviors of users. We observe that users are motivated to post new articles if there is a retweeting mechanism. Furthermore, agents in dense networks are motivated to comment on the articles posted by others because articles spread widely among users, and thus, users can be incentivized to post articles.

Keywords: Social media · Agent-based simulation · Meta-norms game · Complex networks · Retweeting

1 Introduction

In recent years, many social media platforms, including Twitter, Facebook, and Instagram, have drawn significant attention from people around the world. Countless people constantly use these platforms to submit various types of information such as texts, images, and videos for different purposes including personal/group communication, business [6], education [4], and political matters [5]. This collection of information has become a valuable resource/asset shared by social media users. To further grow these assets, we must determine the factor that motivates and facilitates people to provide information, as articles and associated comments must be continuously updated by users.

This issue has been studied from different viewpoints including social psychology [3, 7, 21], social network analysis [23], and agent-based simulation with evolutionary game theory [8, 9, 12, 15]. Zhao et al. [23], for example, reported the potential impacts of micro-blogging sites, such as Twitter. They also attempted to understand the reason behind people using micro-blogs as an informal communicative tool and the user behavior features. Toriumi et al. [15] modeled the activities of social media by modifying the public goods game [1] as posted articles are shared resources. However, because people have to incur some costs and responsibilities by posting articles, they may become lurkers, who just read articles without posting any. They introduced the following: 1) rewards, which correspond to writing comments on posted articles, 2) cooperation, which corresponds to posting new articles, and 3) meta-rewards, which correspond to a comment made on an existing comment; they showed that meta-rewards enhance cooperation [15]. However, the effect of *retweeting* on the activities of social media users has not been studied thus far, although it is evident that retweeting prompts information dissemination and thus increases the motivation for cooperation, i.e., increases the number of posting activities.

Retweeting, a mechanism implemented by a few social media platforms, enables users to read the articles posted by strangers (who are within a user's social network connections) and present their opinions as a reply to the article writers. Consequently, the number of potential readers/commenters of the posted articles may significantly increase. Additionally, we think that retweeting enhances the importance of micro-blogging/tweeting while incurring only a small cost. Thus, investigating the influence of retweeting can help understand the conditions required for ensuring the lasting impact of social media.

Therefore, we extend a reward game model by introducing the *retweeting* mechanism, to clarify the effect of retweeting on user behavior. Subsequently, we experimentally analyze the effect of retweeting on user communication by using variable parameters, which restrict the spread of retweeted information. For the analysis, we perform an agent-based simulation using a genetic algorithm on networks based on a complete graph and *connecting nearest neighbor (CNN) model* [19]. Our results indicate that the probability that a user will retweet an article is moderate, and this enhances cooperative activities (i.e., posting articles) among users, although a reward game without the retweeting mechanism cannot maintain cooperation because of the lack of meta-rewards. Additionally, we observed that agents tend to comment more on dense networks.

2 Related Works

Several studies have been conducted to clarify the role of retweeting; however, most studies aimed to analyze the contents of retweeting or understand the behaviors of users from a psychological viewpoint. Boyd et al. [2] investigated retweeting from a conversational viewpoint and systematically analyzed the syntax of retweets and tried to understand why, how, and what Twitter users retweet. Suh et al. [13] also performed a mathematical analysis to clarify the

factors associated with the retweet rate and built a predictive retweet model for further analysis. They found that URLs, hash tags, and the numbers of followers and followees mostly affect *retweetability*, which is the number of times a tweet can be retweeted. Yang et al. [20] proposed a *factor graph model* to study the underlying mechanism of the retweeting behaviors of users. Zhang et al. [22] defined the notion of social influence locality and predicted the retweeting behaviors of users by training a logistic regression classifier. Ten Thij et al. [14] designed a mathematical model to describe the evolution of a retweet graph. Other studies attempted to identify the typical behaviors of users who frequently retweet. By focusing on a specific user whose profile was known, Luo et al. [10] aimed to investigate the type of followers who tend to frequently retweet the tweets of that specific user. However, these studies focused on the factors that influence retweeting behaviors, and few studies investigated the manner in which the presence of the retweeting mechanism influences the behaviors of users toward posting new articles and commenting on them.

In addition to Toriumi et al. [15], several studies further investigated a model that is an extension of the public goods game mentioned in Sect. 1 [15], to clarify the conditions for ensuring the lasting impact of social media. Hirahara et al. [8,9] extended this model by adding low-cost feedbacks such as the “Like!” button and “read marks” feature. Despite not having a meta-reward mechanism, they considerably facilitated cooperation through an agent-based simulation that was executed on complex networks generated using Facebook data. Osaka et al. [12] extended the model by introducing *direct reciprocity* into it, and they studied the effect of direct reciprocity and network structure on the continuing prosperity of social networking services. Toriumi et al. [16] further explored what types of incentive systems of rewards and punishments promote and maintain effective cooperation in actual groupware. Toriumi et al. [17,18] updated the meta-reward model to identify a realistic situation through which to achieve a cooperation in *Consumer-Generated Media* and analyzed the effects of the information behaviors. However, they did not focus on how the existence of the retweeting mechanism influences user strategies. Therefore, we propose an evolutionary model based on a reward game to explore how the willingness of users toward posting articles and commenting would vary in social media depending on the presence of a retweeting mechanism.

3 Proposed Model

3.1 Reward Game with Retweeting

To model user behaviors, including retweeting, observed in social media, we propose a *retweet (RT) reward game*, which is an evolutionary game based on networked agents. The game is an extension of the *reward game* proposed by Toriumi et al. [15]. The game is extended by adding several rounds of retweeting for each article posted.

Intuitively, retweeting is the behavior of re-posting or forwarding the tweets of a person to her/his followers. Posting an article is often called *cooperating*,

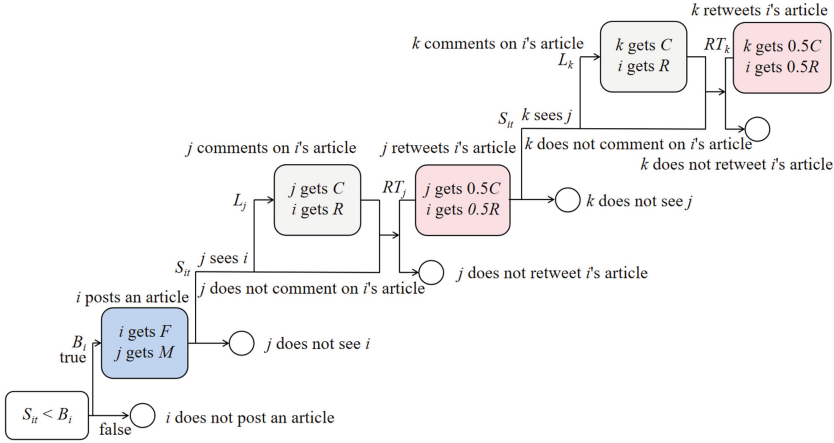


Fig. 1. RT reward game.

and commenting on an article is often called *rewarding*. Let A denote a set of n agents that correspond to users in a social networking service. The agent network is denoted by graph $G = (A, E)$, where E signifies the set of undirected edges representing friendly (i.e., followee/follower) relationships. Because we assume the edges to be undirected, all the agents are mutually connected, i.e., users involuntarily follow back their followers, for simplicity. N_i denotes the set of neighbors of agent i , which lies in G , i.e., $\forall i \in A$

$$N_i = \{j \in A | e_{ij} \in E\},$$

where e_{ij} denotes the edge between agents i and j . Agent i has three learning parameters that decide his/her behavior: *cooperation rate* B_i , *comment rate* L_i , and *retweeting rate* RT_i ; the values of these parameters describe the probabilities of the corresponding behavior.

The RT reward game proceeds as follows (see Fig. 1). Parameter S_{it} ($0 \leq S_{it} \leq 1$), which is called the *seeing probability*, indicates how interesting the article of agent i is at time t and is randomly decided by the game environment. For agent i , if $S_{it} < B_i$, then agent i cooperates (i.e., i posts an article or tweets with probability B_i). If agent i cooperates, all the agents in N_i receive a positive reward M , and agent i receives a negative reward F (so cost) for posting the article. Agent $j \in N_i$ views the article of agent i with probability S_{it} . If agent j views the article of agent i , then agent j will comment on the article of agent i with probability L_j . If agent j comments, then agent j receives negative reward C as the cost of writing a comment, and agent i receives a positive reward R . As long as agent j views the article of agent i , agent j may retweet the article of agent i with probability RT_j . If agent j retweets the article of agent i , agent j receives $0.5C$, and agent i receives $0.5R$. Agent $\forall k \in N_j$ has a chance to see the article of agent i with probability S_{it} . If agent k views the retweeted article of agent i and has not commented on it before, agent k can comment on the

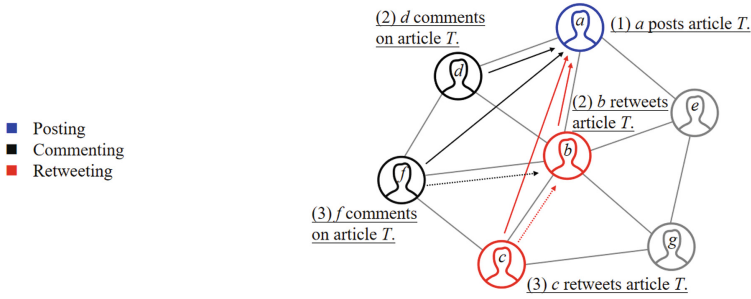


Fig. 2. Flow of tweet: article posting, comment, and retweet.

article with probability L_k . If agent k comments on the article, agent k receives a negative reward C , and agent i receives a positive reward R . Furthermore, if agent k is yet to retweet the article, agent k can retweet it with probability RT_k . If agent k retweets the article, agent k receives $0.5C$, and agent i receives $0.5R$. This ends one period of the RT reward game for agent i .

We set the reward and cost associated with retweeting to half of the values of reward R and cost C , as retweeting is only a tool used to spread articles by forwarding information without including any comment. Additionally, clicking the retweet button should cost less than writing a comment and brings relatively less rewards to the cooperators. Unlike the “Like!” button or “read marks” feature, a retweeted article is considered as content posted by a retweeter and thus influences his/her audience, indicating stronger admiration; therefore, the cooperator should be responsible while disseminating the article. When all the agents in A have ended their respective periods of the RT reward game, one round of the game is completed. The parameters previously mentioned are summarized in Table 1.

Let us consider an example wherein seven users $\{a, b, c, d, e, f, g\}$ are connected in a social network, as shown in Fig. 2. First, suppose that user a posts a new article T (see (1) in Fig. 2). Only her/his friends, d, b , and e , can see T . Second (see (2) in Fig. 2), one of a ’s friends, assume d , reads the article and decides to comment on the article, and another friend b also reads T and decides to retweet it; however, e does nothing after reading T . Because b retweets T , her/his friends, f, c , and g (including a, d , and e) may be able to see T . Because d has previously commented on T , d does nothing; however, e , who has not commented on the article till now, may comment on a ’s article. This implies that an agent who is not a friend of the article poster would have a chance to comment on the article if one of the agent’s neighbors retweets the article. Third (see (3) in Fig. 2), a friend of b , assume f , reads T and comments on it, and c also retweets T again. Notably, a user can simultaneously be a commenter and retweeter of the same article.

Table 1. Parameters

Parameter	Description	Value in Exp
S_{it}	Seeing probability of agent i at period t ,	$0 < S_{it} < 1$
F	Cost of posting an article, $F < 0$	-3.0
M	Reward upon reading an article, $M > 0$	1.0
C	Cost of a comment, $C < 0$	-2.0
R	Reward from posting a comment, $R > 0$	9.0

3.2 Evolutionary Process in Networked Agents

One generation is defined as four rounds of the above-mentioned game. After four rounds, each agent calculates its payoff, which is the total rewards received in the current generation. Payoff is used as the fitness value of an agent for evolution. Notably, the fitness values will be cleared before the beginning of each generation. The parameters that determine the agent behavior, B_i , L_i , and RT_i , are encoded using 3 bits, whose values correspond to $0/7, 1/7, \dots, 7/7$; therefore, in total, the agents have their own 9-bit genes.

The genetic algorithm used in our experiments comprises *parent selection*, *crossover*, and *mutation*. In parent selection, after calculating the fitness values of all the agents, agent i chooses two agents from $N_i \cup \{i\}$ as the parents for the next generation. The probability of j ($\in N_i \cup \{i\}$) being chosen is calculated as follows:

$$\Pi_j = \frac{(v_j - v_{min})^2}{\sum_{k \in N_i \cup \{i\}} (v_k - v_{min})^2}, \tag{1}$$

where v_j denotes the fitness value of j , and v_{min} denotes the minimum fitness value among those of $N_i \cup \{i\}$. Agents with high fitness values are likely to be selected as parents by their neighboring agents.

After choosing two parent agents, agent i generates a new gene through uniform crossover, which implies that each bit of the new gene is chosen from either of the two parent agents with equal probability. After building the new 9-bit gene in the crossover process, each bit is inverted with the probability of 0.01. This probability is called the *mutation rate*. Subsequently, the gene obtained is used as the gene for agent i in the next generation.

4 Experiment

4.1 Experimental Settings

The objective of this experiment was to investigate the dominant strategy that was common among users. The strategy is specified by B_i , L_i , and RT_i when a retweeting mechanism is introduced in an SNS. Conversely, we investigated the most beneficial behaviors when the neighbors also have the same or similar strategies to those of the poster. This dominant strategy also suggests the manner

Table 2. Characteristics of CNN networks (number of agents $n = 1000$)

Parameter	$u = 0.1$	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Average degree	2.20	2.44	2.77	3.19	3.84	4.78	6.40	9.68	17.6
Cluster coefficient	0.39	0.41	0.45	0.44	0.56	0.60	0.68	0.80	0.87

in which the willingness of users toward posting articles and commenting would be promoted or diminished by the retweeting mechanism. These behaviors can be analyzed in terms of the posting rate (or the cooperating rate), B , comment rate (or rewarding rate), L , and retweeting rate, RT , by comparing with a strategy without the retweeting mechanism. Notably, B , L , and RT denote the average parameter values; for example, B is defined as $\sum_{i \in A} B_i / |A|$.

We conducted our experiments using a complete graph and CNN networks, which are based on the CNN model [19], as they are often used in this type of experiments. The number of agents in the complete graph and CNN networks was 20 and 1000, respectively. The characteristics of the CNN networks are presented in Table 2. When the CNN networks were generated, we varied the *probability of changing a potential edge to a real edge*, u , from 0.1 to 0.9.

The other parameter values of the RT reward game are also listed in Table 1. The values are set on the basis of previous studies [1, 15]. All the results in this study are the average values of ten independent runs with different random seeds; however, the results of the complete graph-based experiments are the average values of 100 independent runs.

4.2 Experimental Results - Complete Graph

We chose the reward game and RT reward game to investigate the effect of retweeting on user behaviors. We did not choose the (RT) meta-reward game because we knew that a meta-reward game, in which all the agents have chances to meta-reward, can maintain high cooperating and comment rates because of the effect of posting a comment on another comment (i.e., meta-reward). Thus, it will become difficult to understand the effect of retweeting on the behaviors of users. However, agents that play the reward game on CNN networks and complete graphs have low activities, and thus, changes are easily observed. First, we experimented the reward game and RT reward game on a complete graph. The average cooperating and comment rates versus time are plotted in Fig. 3.

In the reward game, although the average cooperating rate of all the generations was 0.1527 (fairly low), it increased to 0.9060 after introducing retweeting (i.e., the RT reward game). Simultaneously, there was a minor increase in the average comment rate, from 0.0287 to 0.0841. The increase in the comment rate indicates that although the commenting activities were not substantially affected, the retweeting mechanism considerably activated cooperation, i.e., the posting/tweeting behavior. To estimate the extent to which the cooperating rate was changed, we defined the *increasing ratio* of B as follows:

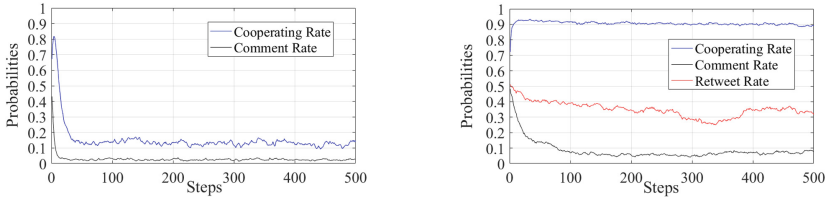


Fig. 3. Averages of the learning parameters on a complete graph.

$$inc = \frac{B_{rt} - B_{normal}}{B_{normal}} \times 100\%, \tag{2}$$

where B_{normal} and B_{rt} denote the average values of cooperating rates in the reward game and RT reward game, respectively. The increasing ratio was 4.93 when the network was a complete graph.

4.3 Experimental Results - CNN Networks

The results of the reward game and RT reward game on CNN networks are shown in Figs. 4 and 5. The cooperating rates, comment rates, and increasing ratios, derived by varying the values of u from 0.1 to 0.9 in steps of 0.1, are listed in Table 3, wherein each element is the average between 300 and 500 generations. Generally, with regard to the cooperating rate, the results obtained on the CNN networks showed similar tendency to those obtained on the complete graph. Conversely, the cooperating rate increased on introducing the retweeting mechanism. The relationship between the increasing ratio and u is plotted in Fig. 6a. From the figure, it is evident that the increasing ratio was higher for u values ranging from 0.1 to 0.7 and lower for u values ranging from 0.7 to 0.9. However, it appears that the comment rate gradually decreased with an increase in u (see Table 3).

The results indicate that the agents are more willing to post new articles after retweeting is implemented. This observation is reasonable because retweeting increases the chances of an article being read by users located at slightly longer distances from the original article poster in the network. First, we focus on the results of the reward game. From Fig. 4, it is indicated that the cooperating rate was approximately 0.5; therefore, cooperation was moderately active unlike the results of the complete graph. The cooperating rate first decreased slightly as u increased from 0.1 to 0.7. It then quickly increased as u increased from 0.7 to 0.9. However, the comment rate, L , constantly decreased with an increase in u . In contrast, as shown in Fig. 5, the cooperating rate in the RT reward game was considerably higher than that in the reward game and constantly increased as u increased. Similar to the reward game, the comment rate, L , decreased in the RT reward game. This comparison is also shown in Fig. 6b. Because the cooperating rate of the reward game was the lowest for $u = 0.7$, as shown in Fig. 6b, the increasing ratio was also maximum when $u = 0.7$. However, the value of RT did not change considerably (see Fig. 5 and Table 3).

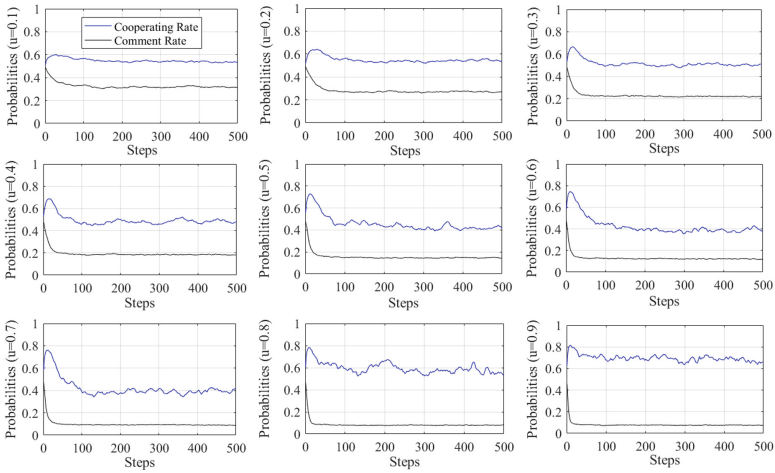


Fig. 4. Cooperating and comment rates of the reward game on CNN networks.

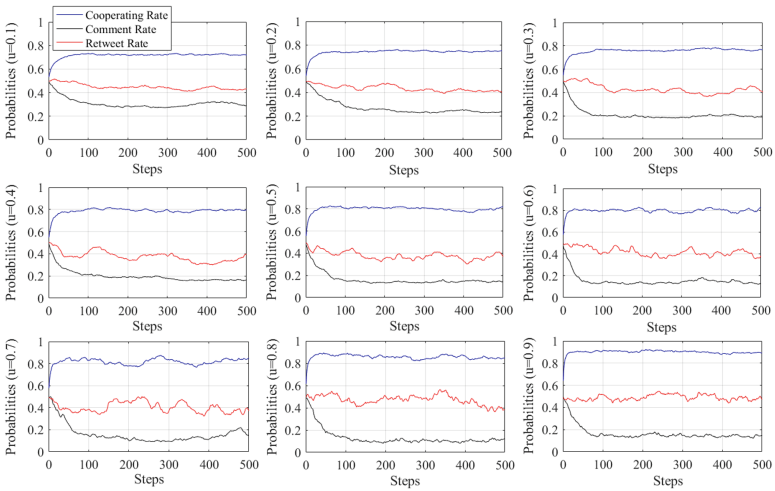
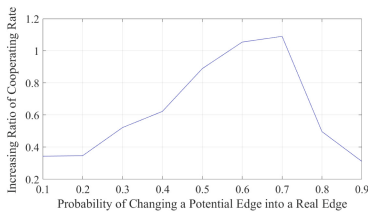
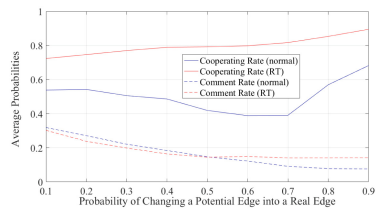


Fig. 5. Cooperating and comment rates of the RT reward game on CNN networks.



Increasing ratio of B .



Cooperating and comment rates.

Fig. 6. Parameter comparison in (RT) reward games.

Table 3. List of cooperating rates, comment rates, retweeting rates, and increasing ratios on various CNN networks.

u	Game model	B	L	RT	Inc. ratio of B
0.1	Reward game	0.5384	0.3187	—	0.3442
	RT reward game	0.7237	0.3024	0.4302	
0.2	Reward game	0.5423	0.2711	—	0.3471
	RT reward game	0.7465	0.2380	0.4140	
0.3	Reward game	0.5058	0.2210	—	0.5224
	RT reward game	0.7701	0.1987	0.4059	
0.4	Reward game	0.4862	0.1843	—	0.6231
	RT reward game	0.7892	0.1640	0.3402	
0.5	Reward game	0.4191	0.1471	—	0.8896
	RT reward game	0.7919	0.1449	0.3688	
0.6	Reward game	0.3884	0.1222	—	1.0547
	RT reward game	0.7982	0.1491	0.4075	
0.7	Reward game	0.3908	0.0913	—	1.0905
	RT reward game	0.8170	0.1407	0.3875	
0.8	Reward game	0.5699	0.0796	—	0.4972
	RT reward game	0.8533	0.1040	0.4572	
0.9	Reward game	0.6823	0.0759	—	0.3117
	RT reward game	0.8950	0.1416	0.4873	

4.4 Discussion

In the case of complete graphs, retweeting enables an agent, who had missed an article in the original post, to read the article. Furthermore, for those agents who have previously read an article but have not done anything to it, retweeting could compel them to reread the article and think twice on whether to do something. Whenever an agent retweets an article, the neighbor agents come to know that other agents are interested in the article and will get a new chance to read it. Thus, they can perform some activities such as commenting or retweeting. Therefore, retweeting in complete graphs considerably increases the chance of an article to be read and commented on. In the case of CNN networks, retweeting may also help activate some friends of the article poster provided that the retweeters and posters have some mutual friends or their friends are friends. Simultaneously, retweeting increases the number of potential readers by allowing strangers to read and act on an article. All these effects make the article posters highly likely to receive comments, thereby making article posting significantly proficient. In CNN networks, the cooperating rate seems to increase the most near $u = 0.7$. We suppose it is because the posting rate of reward game falls from $u = 0.1$ to $u = 0.7$, and rises from $u = 0.7$ to $u = 0.9$.

However, the comment rate also increases in some networks, which would imply that agents become willing to comment on the articles of others. Those who comment more should lose more fitness value. After retweeting is implemented, the posting rate increases with a subsequent increase in the chance of reading; therefore, agents will have more chances to choose whether to reward others, and thus, commenters who are less active may also benefit. The results demonstrate that agents in networks with a high u are dense and will have increased comment rates after the implementation of retweeting. A complete graph is an extreme case of a dense network, and the cooperating rate in it is fairly high.

5 Conclusion

We investigated the effect of retweeting on social media users. We extended the reward game by introducing the retweeting mechanism. In the new model, each article undergoes two rounds of retweeting. The new retweeting mechanism allows users to read the articles of strangers, thereby increasing the number of potential readers for article posters. We analyzed the manner in which the posting and comment rates of the agents would change upon the implementation of retweeting. We found that retweeting motivates agents to post new articles. In CNN networks, the cooperating rate seems to increase the most for u values near 0.7.

In the future, we plan to implement meta-rewards in our model, run an agent-based simulation on real networks like Facebook, and apply the *multiple world genetic algorithm* [11] to analyze the diversity of agent strategies.

Acknowledgements. This work is partly supported by JSPS KAKENHI Grant Number 20H04245, 19H02376, 18H03498 and 17KT0044. We thank the Program Committee for their insightful comments.

References

1. Axelrod, R.: An evolutionary approach to norms. *Am. Polit. Sci. Rev.* **80**, 1095–1111 (1986)
2. Boyd, D., Golder, S., Lotan, G.: Tweet, tweet, retweet: conversational aspects of retweeting on Twitter. In: 2010 43rd Hawaii International Conference on System Sciences, pp. 1–10. IEEE (2010)
3. Burke, M., Marlow, C., Lento, T.: Social network activity and social well-being. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1909–1912 (2010)
4. Chugh, R., Ruhi, U.: Social media in higher education: a literature review of Facebook. *Educ. Inf. Technol.* **23**(2), 605–616 (2018)
5. Conway, B.A., Kenski, K., Wang, D.: The rise of Twitter in the political campaign: searching for intermedia agenda-setting effects in the presidential primary. *J. Comput.-Mediated Commun.* **20**(4), 363–380 (2015)
6. Culnan, M.J., McHugh, P.J., Zubillaga, J.I.: How large us companies can use Twitter and other social media to gain business value. *MIS Q. Executive* **9**(4) (2010)

7. Ellison, N., Steinfield, C., Lampe, C.: Spatially bounded online social networks and social capital. *Int. Commun. Assoc.* **36**(1-37) (2006)
8. Hirahara, Y., Toriumi, F., Sugawara, T.: Evolution of cooperation in SNS-norms game on complex networks and real social networks. In: *International Conference on Social Informatics*, pp. 112–120. Springer, Heidelberg (2014)
9. Hirahara, Y., Toriumi, F., Sugawara, T.: Cooperation-dominant situations in SNS-norms game on complex and Facebook networks. *New Gen. Comput.* **34**(3), 273–290 (2016)
10. Luo, Z., Osborne, M., Tang, J., Wang, T.: Who will retweet me? Finding retweeters in Twitter. In: *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 869–872 (2013)
11. Miura, Y., Toriumi, F., Sugawara, T.: Multiple-world genetic algorithm to identify locally reasonable behaviors in complex social networks. In: *2019 IEEE International Conference on Systems, Man and Cybernetics*, pp. 3665–3672 (2019)
12. Osaka, K., Toriumi, F., Sugawara, T.: Effect of direct reciprocity and network structure on continuing prosperity of social networking services. *Comput. Soc. Netw.* **4**(1), 2 (2017)
13. Suh, B., Hong, L., Pirolli, P., Chi, E.H.: Want to be retweeted? Large scale analytics on factors impacting retweet in Twitter network. In: *2010 IEEE Second International Conference on Social Computing*, pp. 177–184. IEEE (2010)
14. Ten Thij, M., Ouboter, T., Worm, D., Litvak, N., van den Berg, H., Bhulai, S.: Modelling of trends in Twitter using retweet graph dynamics. In: *International Workshop on Algorithms and Models for the Web-Graph*, pp. 132–147. Springer, Heidelberg (2014)
15. Toriumi, F., Yamamoto, H., Okada, I.: Why do people use social media? Agent-based simulation and population dynamics analysis of the evolution of cooperation in social media. In: *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, vol. 2, pp. 43–50. IEEE (2012)
16. Toriumi, F., Yamamoto, H., Okada, I.: Exploring an effective incentive system on a groupware. *J. Artif. Soc. Soc. Simul.* **19**(4), 6 (2016). <https://doi.org/10.18564/jasss.3166>
17. Toriumi, F., Yamamoto, H., Okada, I.: A belief in rewards accelerates cooperation on consumer-generated media. *J. Comput. Soc. Sci.* **3**, 19–31 (2019)
18. Toriumi, F., Yamamoto, H., Okada, I.: Rewards visualization system promotes information provision. In: *Annual Conference of the Japanese Society for Artificial Intelligence*, pp. 55–65. Springer, Heidelberg (2019)
19. Vázquez, A.: Growing network with local rules: preferential attachment, clustering hierarchy, and degree correlations. *Phys. Rev. E* **67**(5), 056,104 (2003)
20. Yang, Z., Guo, J., Cai, K., Tang, J., Li, J., Zhang, L., Su, Z.: Understanding retweeting behaviors in social networks. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pp. 1633–1636 (2010)
21. Yu, A.Y., Tian, S.W., Vogel, D., Kwok, R.C.W.: Can learning be virtually boosted? An investigation of online social networking impacts. *Comput. Educ.* **55**(4), 1494–1503 (2010)
22. Zhang, J., Liu, B., Tang, J., Chen, T., Li, J.: Social influence locality for modeling retweeting behaviors. *IJCAI* **13**, 2761–2767 (2013)
23. Zhao, D., Rosson, M.B.: How and why people Twitter: the role that micro-blogging plays in informal communication at work. In: *Proceedings of the ACM 2009 International Conference on Supporting Group Work*, pp. 243–252 (2009)

Author Index

A

Almeida-Ñauñay, Andrés F., 609
Amuda, R., 533
Asatani, Kimitaka, 308
Atzmueller, Martin, 249
Aynulin, Rinat, 27

B

Ba, Cheick Tidiane, 581
Backenköhler, Michael, 432
Baltsou, Georgia, 164
BarNoy, Amotz, 319
Basov, Nikita, 330
Benito, Rosa M., 494, 609
Bera, Debajyoti, 112
Bloemheuvel, Stefan, 249
Bochenina, Klavdiya, 100
Bouvry, Pascal, 200
Brede, Markus, 382
Brust, Matthias R., 200

C

Cai, Zhongqi, 382
Carscadden, Henry L., 455
Cazabet, Remy, 522
Chandrasekar, V. K., 533
Chebotarev, Pavel, 27
Chen, Jialu, 237
Chepovskiy, A. A., 38
Cherifi, Hocine, 211, 284
Chunaev, Petr, 100
Cote, Shana, 225

D

Danoy, Gregoire, 200
Dhama, Sakshi, 137
Díaz, Robert, 225
Dilmaghani, Saharnaz, 200
Dombi, József, 137
Doran, Derek, 296

E

Eades, Peter, 237
El Hassouni, Mohammed, 284

F

Fushimi, Takayasu, 469

G

Gabrys, Bogdan, 62
Gaito, Sabrina, 581
Galeano, Javier, 494
Gal-Oz, Nurit, 641
Gerding, Enrico, 382
Gounaris, Anastasios, 164
Gradov, Timofey, 100
Großmann, Gerrit, 432
Grubb, Jacob, 357
Gudes, Ehud, 641
Guleva, Valentina Y., 482
Gusrialdi, Azwirman, 509

H

Hammerschmidt, Dennis, 177
Hecking, Tobias, 408

Hirahara, Kazuro, 627
 Hisano, Ryohei, 75
 Hong, Seok-Hee, 237
 Hu, Jingming, 237

I

Inafuku, Kazufumi, 469
 Ito, Mariko I., 273

J

Jia, Mingshan, 62

K

Kadariya, Dipesh, 296
 Kamiński, Bogumił, 152
 Khaykova, S. P., 38
 Klesen, Jonas, 432
 Korkmaz, Gizem, 395
 Krukowski, Simon, 408
 Kuhlman, Chris J., 395, 455

L

Lafhel, Majda, 284
 Lakshmanan, M., 533
 Latapy, Matthieu, 568
 Lavrač, Nada, 420
 Lawryshyn, Yuri, 189
 Leclercq, Eric, 211
 Leshchev, D. A., 38
 Lopez, Derek, 357
 Losada, Juan C., 494, 609

M

Ma, Kwan-Liu, 237
 Magnien, Clémence, 568
 Malick, Rauf Ahmed Shams, 124
 Marathe, Madhav V., 455, 544
 Marbach, Peter, 15
 Marbukh, Vladimir, 556
 Matta, John, 357
 Maulana, Ardian, 651, 660
 Medeuov, Darkhan, 330
 Meyer, Cosima, 177
 Mežnar, Sebastian, 420
 Miasnikof, Pierre, 189
 Migler, Theresa, 262
 Milli, Letizia, 370
 Mirkin, Boris, 3
 Miura, Takahiro, 308
 Mizuno, Takayuki, 75
 Mohan, Bhuvaneshwar, 357

Moreira, Andrés, 593
 Mourchid, Youssef, 284
 Musial, Katarzyna, 62

N

Nakamichi, Lauren, 262
 Narayan, Onuttom, 556
 Novick, Yitzchak, 319

O

Ohnishi, Takaaki, 75, 273
 Orellana, Sebastián, 593

P

Palmer, William R., 87
 Papadopoulos, Apostolos N., 164
 Pitsoulis, Leonidas, 189
 Ponomarenko, Alexander, 189
 Prałat, Paweł, 152
 Premalatha, K., 533
 Priest, Joshua D., 544
 Puzyreva, Ksenia, 330

Q

Quemada, Miguel, 609

R

Rajeh, Stephany, 211
 Rannou, Léo, 568
 Ravi, S. S., 395, 455, 544
 Rebollo, Miguel, 494
 Renoust, Benjamin, 284
 Rivas-Tabares, David, 620
 Rosenkrantz, Daniel J., 455, 544
 Rossetti, Giulio, 370
 Rossi, Gian Paolo, 581
 Roth, Camille, 330

S

Sadeghi, Reza, 296
 Saito, Kazumi, 627
 Sakata, Ichiro, 308
 Saniee, Iraj, 556
 Satoh, Tetsuji, 469
 Savonnet, Marinette, 211
 Senthilvelan, M., 533
 Shalileh, Soroosh, 3
 Sheikh, Maham Mobin, 124
 Shestopaloff, Alexander Y., 189
 Situngkir, Hokky, 651, 660
 Škrli, Blaž, 420

Stearns, Richard E., [544](#)
Sugawara, Toshiharu, [671](#)

T

Tarquis, Ana M., [609](#), [620](#)
Théberge, François, [152](#)
Toccaceli, Cecilia, [370](#)
Topîrceanu, Alexandru, [345](#)
Toriumi, Fujio, [671](#)
Torkel, Marnijati, [237](#)
Tsihlias, Konstantinos, [164](#)
Tsuda, Nako, [51](#)
Tsugawa, Sho, [51](#)

U

Ueda, Naonori, [627](#)

V

Vaganov, Danila A., [482](#)
van den Hoogen, Jurgen, [249](#)

Vega-Redondo, Fernando, [395](#)
Voloeh, Nadav, [641](#)

W

Wang, Huijuan, [444](#)
Wolf, Verena, [432](#)
Wood, Zoë, [262](#)

Y

Yamagishi, Yuki, [627](#)
Yan, Yizhou, [671](#)

Z

Zabihimayvan, Mahdieh, [296](#)
Zaykov, Alexey L., [482](#)
Zhang, Larry Yueli, [15](#)
Zhang, Wenning, [75](#)
Zhao, Xunyi, [444](#)
Zheng, Tian, [87](#)
Zignani, Matteo, [581](#)
Zinoviev, Dmitry, [225](#)